

INPUT
Publicación práctica para usuarios de MSX
Revista mensual 1987
Año 1-Número 9 Precio 375 Ptas.
MSX
MUSICA Y SIMULACION DE PARTITURAS
GEN-MON: EL ENSAMBLADOR/ DESENSAMBLADOR MAS ESTANDAR
DISEÑA CON TU RATON
Participa en el concurso de aplicaciones



AÑO 1 NUMERO 9

DIRECTOR: Manuel Pérez
COORDINADOR EDITORIAL: Francisco de Molina
DIRECTOR DE ARTE: Luis F. Balaguer
REALIZACION GRAFICA: Diego Tudela
COLABORADORES: Julio García, Equipo Molisoft, Javier Ferrer, Francisco J. de la Fuente, Antonio Pliego, Ramón Rabasó, Jaime Mardones, Irene Alcaraz
FOTOGRAFIA: Joan Boada

INPUT MSX es una publicación de PLANETA-DE AGOSTINI, S.A.
GERENTE DIVISION DE REVISTAS: Sebastián Martínez

PUBLICIDAD: José Real-Grupo Jota
Madrid: c./ Gral Varela, 35, 3.º-11
Teléf. 270 47 02/03
Barcelona: Avda. de Sarriá, 11-13, 1.º
Teléf. 250 23 99

FOTOMECANICA: TECFA, S.A.

IMPRESION: Sirven Grafic
C/ Gran Vía, 754-756, 08013 Barcelona
Depósito legal: B. 38.113-1986

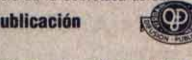
SUSCRIPCIONES: EDISA,
López de Hoyos, 141. 28002 Madrid
Teléf. (91) 415 97 12

REDACCION:
Aribau, 185, 1.º
08021 Barcelona

DISTRIBUIDORA:
R.B.A. PROMOTORA DE EDICIONES, S.A.
Travesera de Gracia, 56. Edificio Odiseus.
08006 Barcelona.

El precio será el mismo para Canarias que para la Península y en él irá incluida la sobretasa aérea.

INPUT MSX es una publicación controlada por





SUMARIO

EDITORIAL 4

APLICACIONES
MUSICA Y SIMULACION DE PARTITURAS 5
PALABRERO 22
MACROIMPRESION 39

PROGRAMACION
DISEÑA CON TU RATON 8

PARTICIPA
LOTO MSX 44

EL MICRO Z-80
EL GRAN IMPACTO DEL MICROPROCESADOR 16

CODIGO MAQUINA
NUMEROS BAJO CERO 48
GEN-MON 51

REVISTA DE SOFTWARE 55

EL ZOCO 66

PROGRAMACION DE JUEGOS (COLECCIONABLE) 31
LA PROGRAMACION PARA JOYSTICK
EL JUEGO DE LA CAZA DE PATOS

SACALE PARTIDO A TU ORDENADOR

Después de la compra del nuevo ordenador o de haber completado tu equipo con nuevos periféricos, ha llegado el momento de que te pongas a trabajar en serio con ellos y deshechar la tentación fácil de aparcarlos en un rincón. Como siempre, INPUT te ofrece la posibilidad de combinar el aprendizaje de las técnicas de programación y la información de las novedades de hardware, con el juego, la enseñanza de trucos útiles y el seguimiento de los más interesantes programas que se publican.

Conscientes de que el mundo de la informática vive un movimiento permanente, del que los medios de comunicación no debemos prescindir, hemos confeccionado una encuesta, con cuyas respuestas esperamos poder sintonizar mejor con vosotros. A fin de que os resulte más sugestiva, se efectuará un sorteo entre todas las contestaciones recibidas.

Por último, hemos de señalar que con la publicación de este número de INPUT inauguramos un nuevo *Concurso de Aplicaciones y Utilidades*, destinado a intentar ampliar la gama de servicios que os prestan vuestros ordenadores.

A nuestro juicio, esta nueva competición estimulará un importante intercambio de experiencias y permitirá introducir nuevos temas en vuestra revista.

Sin duda, vuestras respuestas serán tan entusiastas como en anteriores ocasiones, lo que elevará la calidad y el interés de la revista.

Así pues a DISFRUTAR de vuestro INPUT y... a PARTICIPAR.

LOS MEJORES DE INPUT

Hemos pensado que es interesante disponer de un ***ranking*** que ponga en claro, mes a mes, cuáles son los programas preferidos de nuestros lectores. Para ello, es obligado preguntaros directamente y tener así el mejor termómetro para conocer vuestras preferencias. Podéis votar por cualquier programa aunque no haya sido comentado todavía en **INPUT**.

El resultado de las votaciones será publicado en cada número de **INPUT**.

Entre los votantes sortearemos 10 cintas de los títulos que pidáis en vuestros cupones.

Nota: No es preciso que cortéis la revista, una copia hecha a máquina o una simple fotocopia sirven.

Enviad vuestros votos a: **LOS MEJORES DE INPUT** Aribau, 185. Planta 1. 08021 Barcelona

ELIGE TUS PROGRAMAS

Primer título elegido ____________________

Segundo título elegido ____________________

Tercer título elegido ____________________

Programa que te gustaría conseguir ____________________

Qué ordenador tienes ____________________

Nombre ____________________

1.er Apellido ____________________

2.º Apellido ____________________

Fecha de nacimiento ____________________

Teléfono ____________________

Dirección ____________________

Localidad ____________________

Provincia ____________________

INPUT MSX N.º 9

MUSICA Y SIMULACION DE PARTITURAS

El sistema MSX lleva incorporado un chip que es el encargado de darle el sonido al ordenador, conocido por su gran capacidad. En este artículo te brindamos la oportunidad de iniciarte en la composición musical de la mano de tu ordenador.

Si os fijáis, cuando trabajamos con el microordenador, nos acompañan una serie de sonidos que nos indican funciones; o bien, cuando jugamos y oímos campanas, disparos, canciones y efectos en general. Pues bien, este pequeño programa que presentamos interpreta una melodía que hemos seleccionado.

No es nuestra intención dar una clase de música, para lo que debéis ir al conservatorio, sino divulgar una curiosidad de nuestro ordenador que creemos interesante.

Lo que sí explicaremos es cómo hemos confeccionado el programa, ya que es posible que algún aficionado, además de escuchar la canción que proponemos, quiera hacer un programa para escuchar su composición predilecta.

Para ello es preciso partir de una partitura que debemos codificar, descomponiéndola en notas separadas y silencios, seleccionando según lo completa que queramos la pieza, en melodía principal, primera voz, segunda voz, etcétera.

Una vez estudiada la partitura se trata de codificar cada nota en su letra correspondiente, respetando los silencios, duración de cada nota y octava a que pertenece.

Visto así quizá quede un tanto complejo, pero no lo es, simplemente requiere un poco de atención.

La primera vez es muy posible que la canción no suene como debiera, ya que es fácil olvidar un sostenido o confundir un tiempo, pero... ¡adelante!

Para que podáis ver el desarrollo que hay que seguir, os adjuntamos con el programa un ejemplo de una pieza musical, y la partitura empleada.

Hemos tomado como muestra un fragmento de la Cantata N. 147 de **J. S. Bach,** analizando los 8 primeros compases, que son tal vez la parte más conocida y que define el dibujo musical de la obra.

DESCRIPCION DEL PROGRAMA

En primer lugar hay que definir los parámetros generales: volumen de cada canal y velocidad.

Esta labor se realiza en las instrucciones 190, 200 y 210.

Cada una de estas instrucciones se cuida de un canal del *PSG AY3-8910 (Programable Sound Generator),* que puede controlar hasta tres canales a la vez.

Las variables que hemos utilizado para cada canal son:

R ... para la melodía principal
S ... para la segunda voz
T ... para los bajos o tercera voz.

De la instrucción 220 a 450 vemos los códigos de la pieza interpretados de la siguiente forma: hemos separado la composición en ocho partes, haciendo coincidir cada una con un compás. A su vez, se dividen en tres canales, es decir, para cada compás hay que definir una R, una S y una T.

La instrucción 460 se encarga de programar las variables que caracterizan a la melodía y, por último, desde la instrucción 470 a 550, interpreta los ocho compases secuencialmente.

Podemos modificar la interpretación de nuestra melodía si cambiamos el contenido de los primeros parámetros, creando así nuevas versiones, como por ejemplo:

```
190 RO$="T240V10S1M3000"
```

En la que el canal 1 se ve afectado por una envolvente de una sola vez por nota, y en forma de diente de sierra descendente.

No altera los canales 2 y 3.

Otro caso podría ser:

```
190 RO$="T240V10S14M300"
```

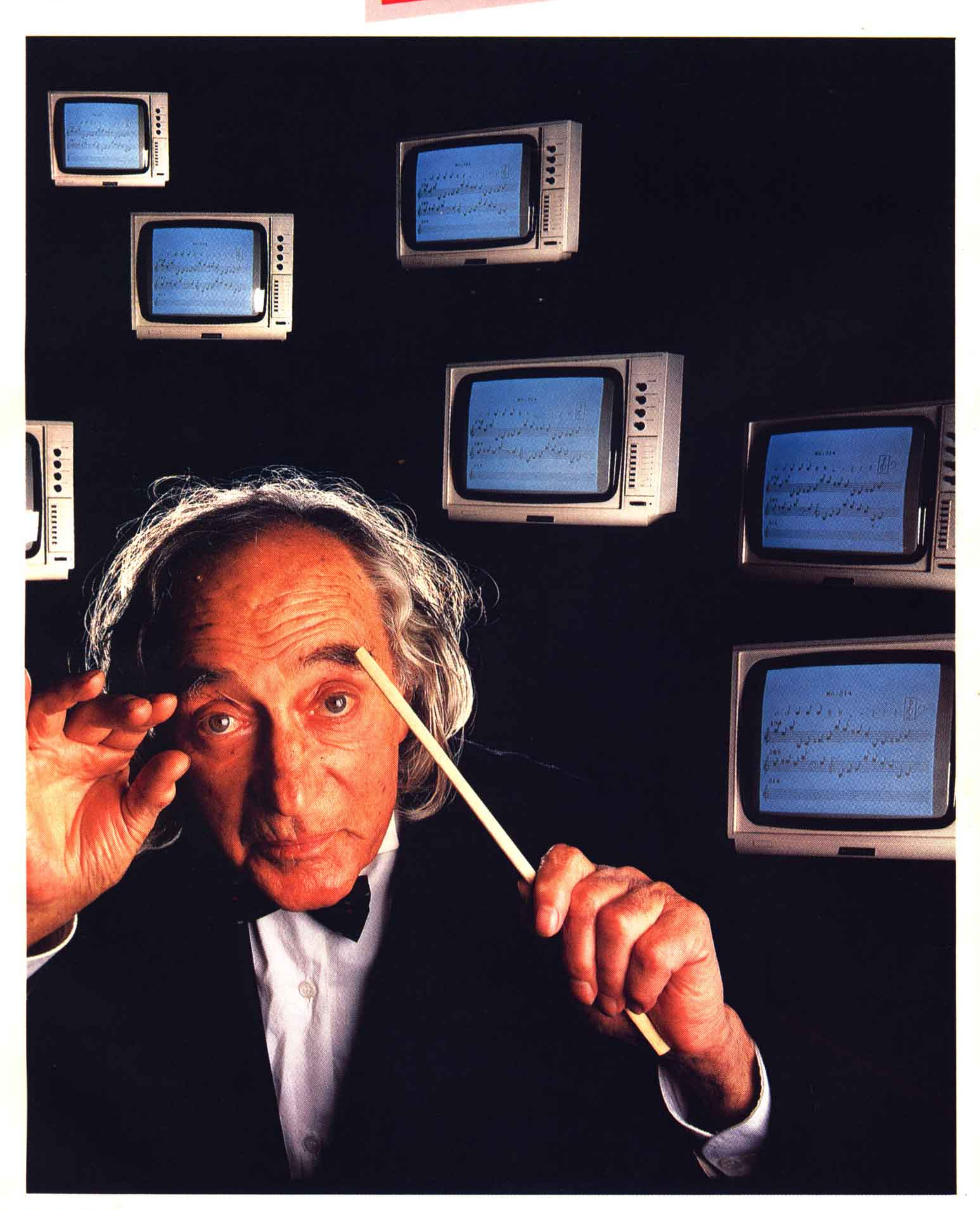
Causa una modulación de baja frecuencia de forma de onda triangular en el canal 1, y no altera los canales 2 y 3.

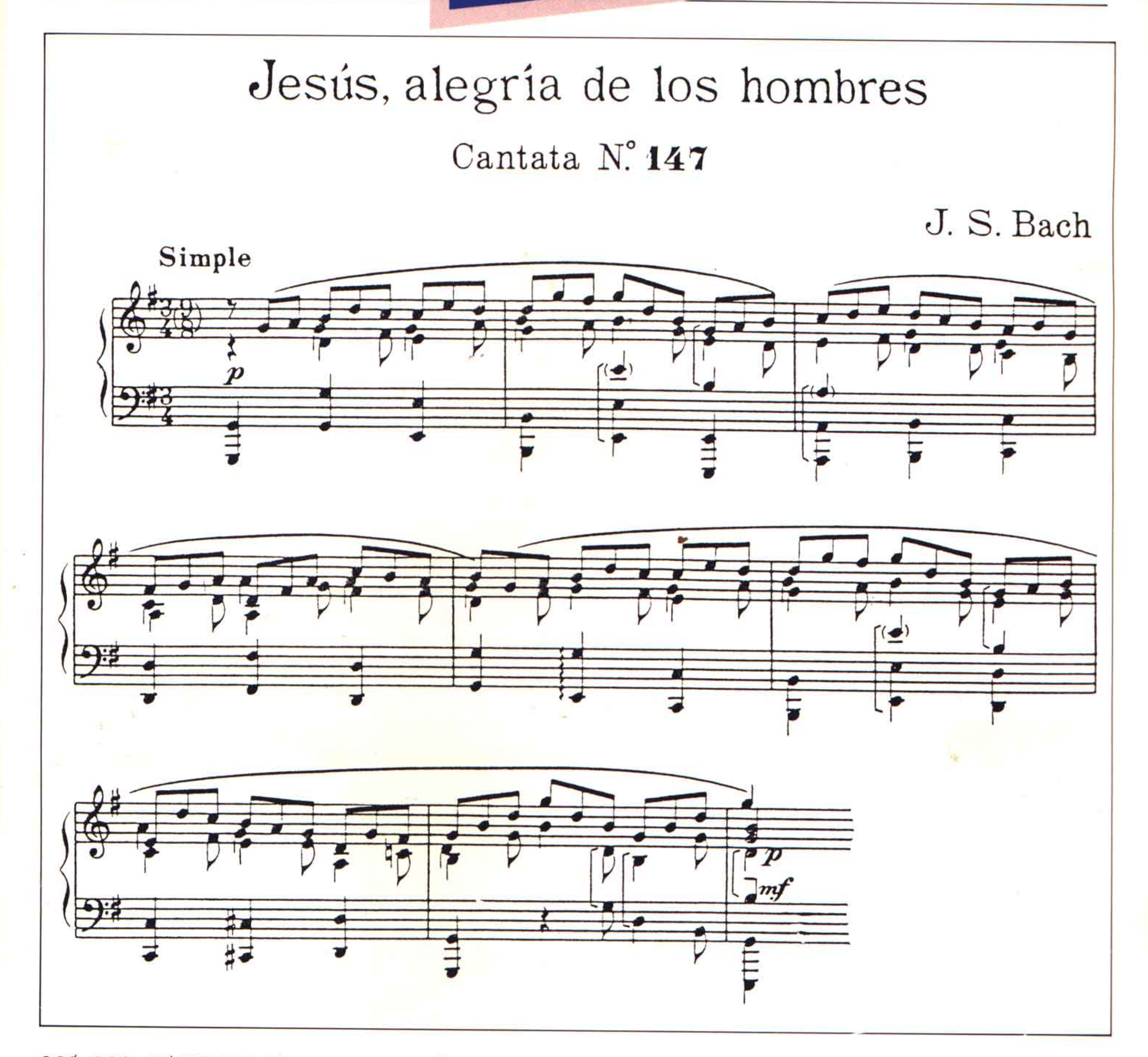
Un consejo importante para cuando codifiquemos nuestras melodías es el verificar que la duración de cada compás sea siempre la misma para todas las voces y en todos los compases, esto puede servirnos de referencia para detectar errores de codificación.

Con lo descrito en este artículo, queremos fomentar la creación de partituras codificadas para MSX entre nuestros lectores.

Aprovechamos la ocasión para informar que en un número próximo de nuestra revista publicaremos las bases de un concurso de canciones y efectos sonoros en el que podéis participar todos vosotros y donde habrán importantes premios para los ganadores. ¡Así que ánimo y... a la canción!

```
100 REM ******************
110 REM *                *
120 REM * Cantata N147
          J.S.Bach.      *
130 REM *                *
140 REM * Por: AY3-8910  *
150 REM *                *
160 REM *                *
170 REM *                *
180 REM ******************
190 R0$="T240V10"
200 S0$="T240V9"
210 T0$="T240V9"
220 R1$="04RGAB05DCCED"
230 S1$="04RRRGGF+GGA"
240 T1$="02GGR03GGREER"
250 R2$="05DGF+GD04BGAB"
```






```
260 S2$="Ø4BBABBGEED"
270 T2$="Ø2BBRØ3EERØ3BB"
280 R3$="Ø5CDEDCØ4BABG"
290 S3$="Ø4EEF+EEDEEØ3B"
300 T3$="Ø2AARBBRØ3CCR"
310 R4$="Ø4F+GADF+AØ5CØ
    4BA"
320 S4$="Ø4CCDØ3AAØ4GAAF
    +"
330 T4$="Ø3DDRF+F+RDDR"
340 R5$="Ø4BGABØ5DCCED"
350 S5$="Ø4GGF+GGF+GGA"
360 T5$="Ø3GGRGGRCCR"
370 R6$="Ø5DGF+GDØ4BGAB
    "
380 S6$="Ø4BBABBGEEG"
390 T6$="Ø2BBRØ3EERØ3DD
    R"
400 R7$="Ø5EDCØ4BAGDGF+
    "
410 S7$="Ø4AAF+GGEØ3AAØ
    4C"
420 T7$="Ø3CCRC+C+RDDR"
430 R8$="Ø4GBØ5DGDØ4BGB
    Ø5DGG"
440 S8$="Ø4DDGBBDØ3BBØ4
    GGG"
450 T8$="Ø2GGRRRGDDØ2BG
    G"
460 PLAYRØ$,SØ$,TØ$
470 PLAYR1$,S1$,T1$
480 PLAY R2$,S2$,T2$
490 PLAY R3$,S3$,T3$
500 PLAY R4$,S4$,T4$
510 PLAY R5$,S5$,T5$
520 PLAY R6$,S6$,T6$
530 PLAY R7$,S7$,T7$
540 PLAY R8$,S8$,T8$
550 END
```

DISEÑA CON TU RATON

Muchos de vosotros disponéis ahora de una valiosa herramienta de trabajo: el ratón. Con el presente artículo pretendemos comentar la importancia que esta utilidad del ordenador supone como avance tecnológico dentro del campo del diseño.

Tras la introducción en el tema, comentaremos un programa que incluimos en este número, con el que podréis realizar pequeños *diseños gráficos*. Lo que sí es muy importante y queremos dejar bien sentado es que en ningún momento hemos pretendido con este programa hacer diseño asistido por ordenador, ya que eso es totalmente distinto, requiere una especialización y unos terminales gráficos con una memoria y prestaciones muy superiores a la de nuestro microordenador.

Cuando hablamos de diseño asistido por ordenador o fabricación asistida por ordenador (CAD/CAM), estamos hablando de ciertas técnicas que aunque nos parecen muy recientes, se han estado utilizando desde finales de los años 60 en el campo del diseño de tarjetas de circuito impreso y en el de trazado de circuitos integrados.

En 1981 aparecieron las primeras estaciones de trabajo CAE (estaciones de trabajo asistido por ordenador enfocadas a la ingeniería), que contemplaban la automatización de las fases preliminares del diseño electrónico. Como consecuencia surgió el mercado ECAD (Diseño electrónico asistido por ordenador).

Actualmente existen estaciones de trabajo para todo tipo de aplicaciones, es decir, que no queda limitado al campo de la electrónica, sino que se emplea en el diseño mecánico, arquitectura, ingeniería y construcción. Se prevé que hacia el año 1989 los terminales de trabajo tendrán el precio y las dimensiones de un ordenador personal actual.

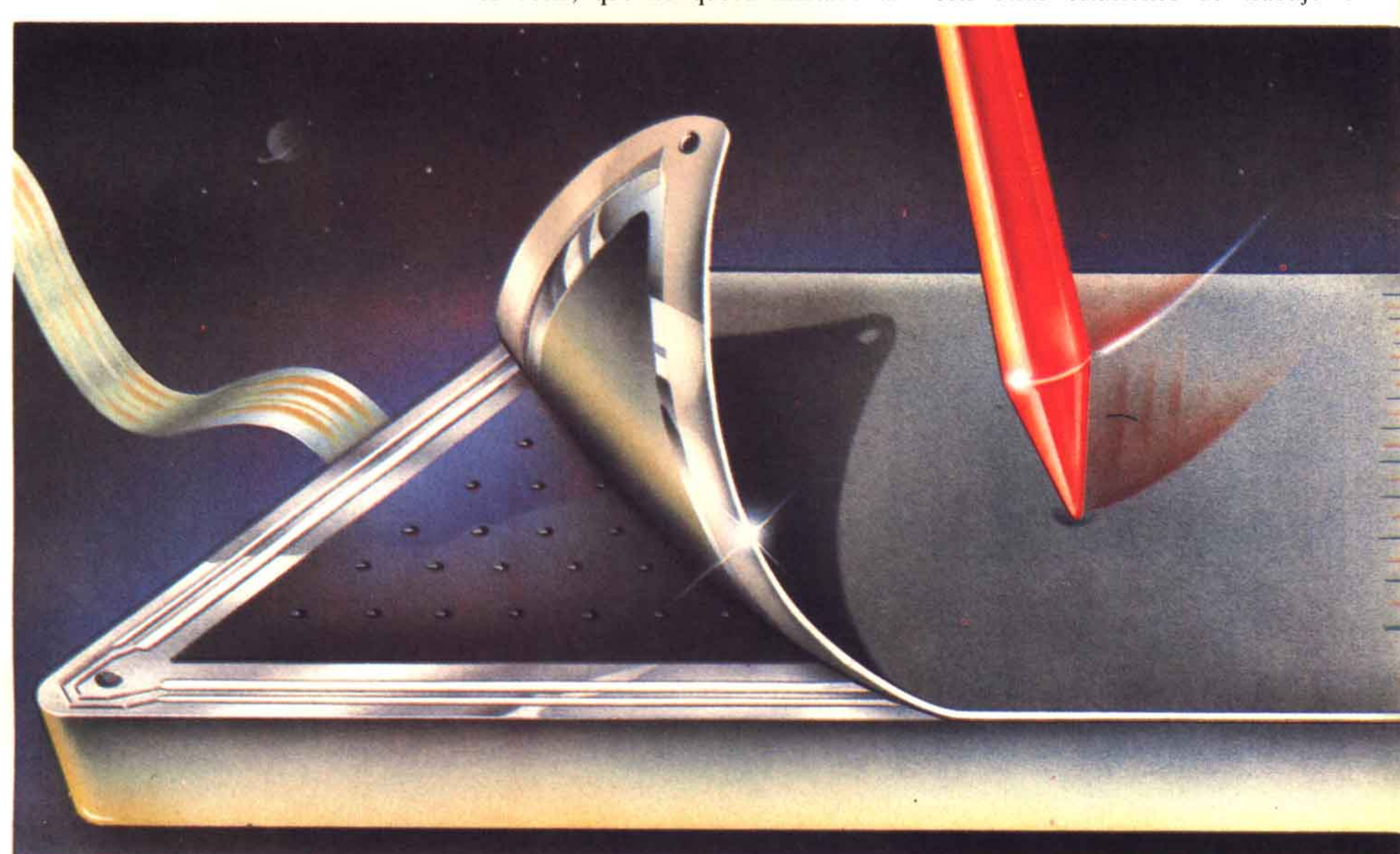
A título informativo os diremos que las especificaciones técnicas mínimas que se precisan para poder realizar aplicaciones de ECAD son:

— Poseer tarjetas controladoras de gráficos B/N y Color con una definición de 1024 * 1024 *pixels* con prestaciones de «*paning*» y «*zoom*».

— Entre 1 y 4 Mbytes de RAM.

— Una capacidad en disco de 40 o 50 Mbytes.

La meta de este sistema de trabajo es conseguir el intercambio de datos con otras estaciones de trabajo o



transferir ficheros a ordenadores más potentes, compartiendo periféricos y bases de datos a través de redes de comunicación locales.

La idea se centra pues en conectar al ordenador central, que se halla en la empresa, las diferentes estaciones de diseño además de los terminales normales de gestión ya existentes.

Los ordenadores de diseño funcionan de una forma interactiva, logrando una comunicación bidireccional hombre-máquina de una forma cómoda y rápida, mediante el lápiz luminoso (*light-pen*), o bien, empleando la denominada tableta gráfica, la cual está más extendida.

Las tabletas gráficas disponen de un apartado o ventana donde se pueden especificar puntos concretos, y seleccionar zonas del dibujo, en las que se desea realizar una función.

En otra parte de la tableta se muestran gran cantidad de funciones predefinidas, como, por ejemplo, trazar un círculo, dado el centro y el radio, trazar círculos que pasen por tres puntos dados, diversas tangentes, efectos de rotación y simetrías, dibujo de rectángulos, cónicas, etc., que permiten aligerar el tiempo de realización del diseño. Asimismo contempla una serie de prestaciones que pueden ser definidas por el usuario, que le van a permitir personalizar la máquina en función de su trabajo.

Los diseños realizados quedan retenidos en el disco, y pueden ser llamados de nuevo para hacer servir parte de los mismos.

Una ventaja importantísima de este sistema de trabajo se centra en la comunicación. En todo momento se puede conocer el estado de cualquiera de las fases del proyecto, llevadas por distintos colaboradores, debido a que funcionan en paralelo y están unidas por una red de comunicación.

EL PROGRAMA

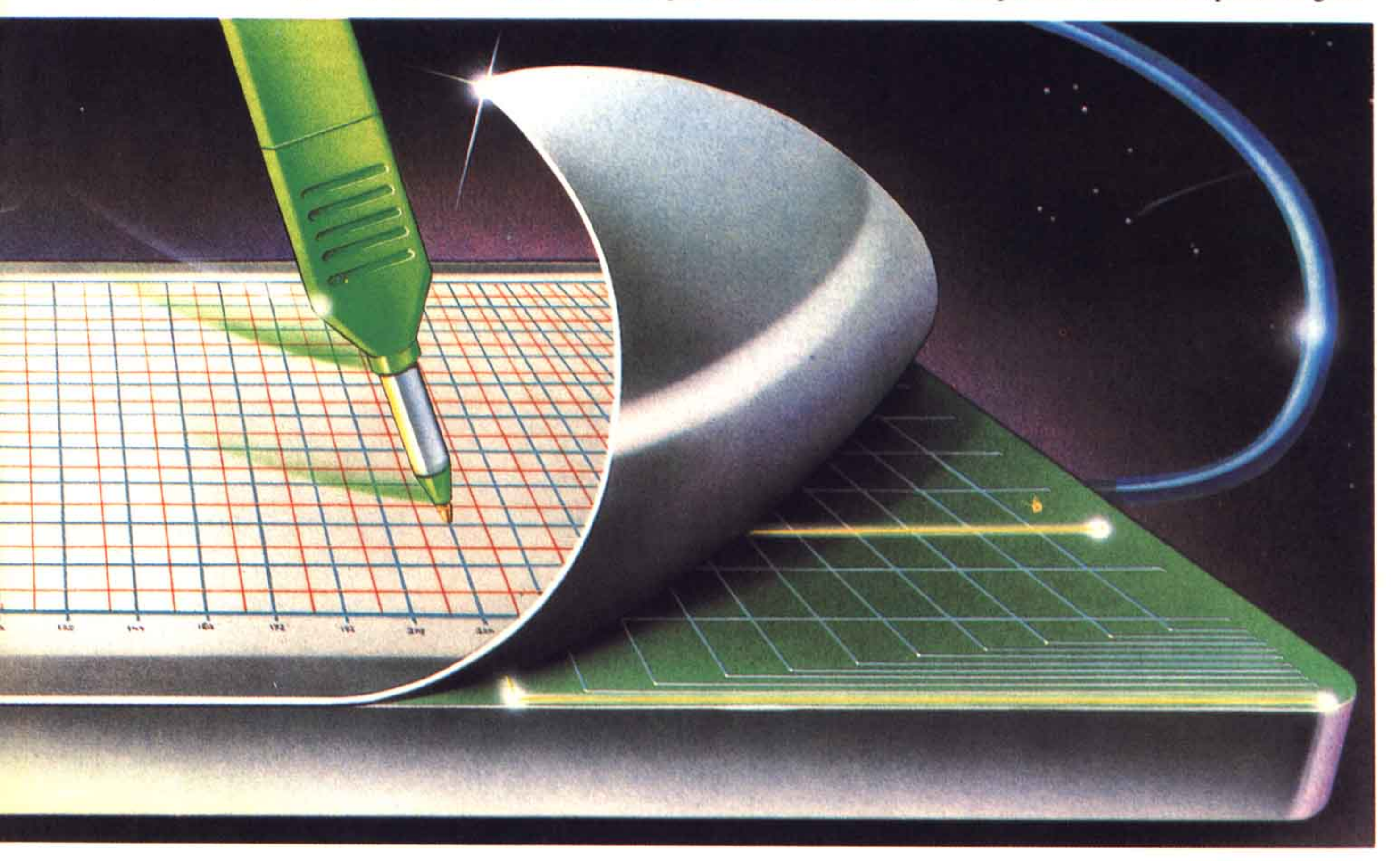
Tal y como citábamos al comienzo de este artículo, el programa presentado es un sencillo ejemplo, y dista mucho de lo que sería un programa de diseño asistido por ordenador, si bien puede satisfacer muchas de tus necesidades sí ansías disponer de un programa que te permita dibujar a tu antojo en la pantalla, sin verte obligada a realizar cálculos para obtener resultados gráficos.

Este pequeño programa permite dibujar trazos en la pantalla gráfica del sistema, de una forma asimétrica y desenfadada. Con esto abrimos una puerta a la creación y al ingenio del lector que desee definir motivos o carátulas que servirán de presentación a sus programas, juegos, o, incluso para las presentaciones de sus películas de video doméstico.

El programa presenta una pantalla a modo de página para dibujar, en la cual podemos seleccionar el color del borde, el color del papel de fondo, y, por último, el color de la tinta a emplear.

El programa puede trabajar hasta con 4 pantallas (0, 1, 2, 3) simultáneamente de 256 × 212 *pixels* y 16 colores.

Una vez tengamos las pantallas completas, tenemos la opción de guar-



darlas en el disco para emplearlas desde otros programas.

Si queremos revisar un dibujo ya almacenado en el disco, lo podemos cargar en la pantalla 0. Incluso una vez cargado podemos modificarlo, cambiar los colores del borde y del papel, y guardarlo de nuevo.

Para dibujar emplearemos el *ratón* del sistema, que adquiere la personalidad de un lápiz o puntero.

Al desplazarlo por encima de la mesa, vemos correr su imagen por la pantalla a modo de cursor. Si queremos que deje rastro por donde pasa, o sea, que dibuje, debemos apretar el pulsador de la izquierda del mismo.

Es muy fácil que deseemos borrar alguno de los trazos realizados. Para ello disponemos de dos gomas, una a nivel de *pixel*, que borra por el simple hecho de volver a pasar el *ratón* por encima del dibujo, y otra a nivel de carácter, que funciona al desplazar el cursor manteniendo apretada la barra espaciadora.

Para cambiar el color del papel, lo podemos hacer de dos formas. El color inicial lo definiremos con ^H (hoja), indicándole a continuación un número del 0 al 7 de la paleta (+,−) previamente seleccionada, o bien, ^P (papel) y el color a continuación. Esta segunda función respeta el dibujo existente sin alterarlo, pero es mucho más lenta en su ejecución, debido a que debe consultar el color de cada punto antes de alterarlo.

Otra particularidad es que nos permite seleccionar hasta 16 colores simultáneamente dentro de la misma pantalla.

Para cambiar de color debemos pulsar ^C y el color deseado.

Antes de emplear algún comando de selección de color debemos seleccionar la paleta a que nos vamos a referir. Existen dos paletas de ocho colores cada una que describimos a continuación:

Código	Paleta baja (−)	Paleta alta (+)
0	Transparente	Rojo
1	Negro	Rojo claro
2	Verde	Amarillo oscuro
3	Verde claro	Amarillo claro
4	Azul oscuro	Verde oscuro
5	Azul claro	Magenta
6	Rojo oscuro	Gris
7	Azul cielo	Blanco

Disponemos de otra opción que es la de rellenar masas de color, es decir, una vez disponemos de una línea lí-



mite cerrada (prestar atención a que no se salga la tinta), debemos situar el cursor dentro de la zona a rellenar y entrar el comando ^F.

Para evitar que al entrar o salir de la figura, el cursor borre la línea límite (permitiendo que al rellenar se salga la tinta), podemos inhibir el cursor con la tecla F2. El cursor volverá a aparecer al pulsar de nuevo esta tecla.

Si seleccionamos el mismo color de papel y tinta, lógicamente el cursor se perdería, pues bien, el programa lo tiene en cuenta y no nos permite poner el color de la tinta igual a la del papel, y viceversa, ni el color del papel igual al de la tinta seleccionada en ese momento.

Por la misma razón, si deseamos borrar todo lo dibujado en un determinado color, podemos cambiar el papel a ese color, volviendo a cambiar después al color deseado. De esta forma el color que queremos borrar quedará absorbido.

Hasta ahora sólo hemos definido las funciones de dibujo simple con el cursor, pero el programa tiene también la posibilidad de dibujar estructuras básicas, como son líneas, rectángulos y círculos. Esta parte la descubrimos a continuación: Si deseamos dibujar una línea, situaremos el cursor en el origen, y teclearemos el comando ^I (inicio). A continuación desplazaremos el *ratón* hasta la posición final y teclearemos el comando ^L (línea). Inmediatamente aparecerá una línea uniendo los dos puntos.

Es de notar que el punto inicial no se pierde hasta definir un nuevo punto inicial, así pues podemos dibujar otra línea desplazando el *ratón*, y tecleando de nuevo ^L (línea).

Por ejemplo, para dibujar un triángulo, el proceso sería el siguiente:

—Llevar el cursor al primer vértice
—Teclear ^I (Inicio)
—Llevar el cursor al segundo vértice
—Teclear ^L (Línea)
—Llevar el cursor al tercer vértice
—Teclear ^L (Línea) y ^I (nuevo Inicio)
—Llevar el cursor al segundo vértice
—Teclear ^L (Línea)

Si, además, lo queremos rellenar, debemos inhibir el cursor (mediante la función F2), observando que el triángulo quede completo y desplazar ligeramente el cursor hacia el interior del triángulo. Pulsar de nuevo F2 para verificar que el cursor se halla en el interior de la figura, y pulsar entonces ^F (Fill=Llenar).

Si queremos trazar figuras poligonales debemos enlazar secuencias ^I ^L, obteniendo líneas encadenadas.

Otra opción que nos brinda este programa es la de dibujar cuadrados dados dos vértices opuestos.

Para hacerlo debemos ir a un vértice y teclear ^I (Inicio).

Seguidamente desplazaremos el cursor hasta el vértice opuesto, y teclearemos ^Q (Cuadrado), e inmediatamente aparecerá un cuadrado en la pantalla.

Como podemos observar, el vértice inicial es el mismo que hemos hecho servir para trazar líneas, por tanto podemos trazar líneas desde el vértice de este cuadrado sin hacer nada especial. Esta función servirá también para encadenar figuras geométricas.

Por último, el programa nos da la posibilidad de dibujar círculos, para ello, debemos situar el cursor en el centro y definirlo como punto inicial (^I) y a continuación, situar el cursor en cualquier punto del círculo a trazar, y teclear seguidamente el comando (^O), inmediatamente aparecerá un círculo, que como en los anteriores casos, se puede rellenar.

Si deseamos círculos concéntricos, no debemos redefinir el centro.

Otra parte del programa es la relacionada con el almacenamiento de los dibujos en el disco. Con tal objeto, disponemos de dos comandos:

La tecla de función (F3) ordena el almacenamiento en disco de la pantalla corriente. Para no modificar el dibujo sobreponiendo caracteres, el mismo programa crea un nombre para este archivo.

El nombre se compone de los caracteres «CD» seguidos de seis cifras, que son las indicadas por la variable TIME y la extensión «.dat» para indicar que se trata de un archivo de datos.

Es muy difícil que en dos sesiones de diseño, aun empleando el mismo disco, se realice una escritura en el mismo instante, pero es recomendable, una vez finalizada la sesión, adjudicar un nombre más apropiado al archivo que contiene el dibujo, empleando el comando NAME del sistema operativo.

Después de guardar el diseño el programa continúa, así que podemos guardar estados intermedios de un dibujo y presentarlos después en secuencia por un programa que se encargue de leerlos del disco y presentarlos en pantalla.

El guardar un dibujo en el disco no altera el contenido de las otras tres pantallas gráficas.

Para visionar dibujos anteriores, el programa dispone de la función (F4) (carga de dibujo) que posibilita pasar un archivo a la pantalla cero (observa que sólo se permite en la pantalla cero).

Debido a que dicho comando destruye la información de esta pantalla, el programa dispone de una protección. Inmediatamente después de pulsar (F4) debemos pulsar la letra «D». Cualquier otra combinación aborta la ejecución de este comando.

Si pulsamos (F4) y «D», aparece en la pantalla el directorio del disco, para poder seleccionar el archivo a cargar.

Cuando lo tenemos seleccionado, basta con pulsar cualquier tecla para que el ordenador nos pregunte por el nombre de dicho fichero. Para evitar errores, el programa toma la precaución de comprobar que el nombre del archivo tenga la extensión «.dat», ya que de lo contrario, ignora el comando.

Si todo es correcto, se cargará en la pantalla cero el dibujo almacenado en el disco (nótese que el cursor pasa a modo inactivo).

Hemos comentado las prestaciones en cuanto a selección y cambio de colores, dibujo simple empleando el *ratón*, dibujos geométricos, rellenado de figuras, y lectura/escritura de diseños en disco. Pero aún nos queda otra posibilidad que es la de escribir texto en la pantalla. Basta con situar el cursor en el lugar deseado y pulsar la tecla

del carácter a escribir. Podemos escoger entre mayúsculas, minúsculas, símbolos y números; sólo están prohibidos los signos + y − que están reservados para cambiar la paleta de colores.

Para finalizar el programa, debemos emplear la tecla (F5), ya que se ha inhibido el stop, destruyendo los contenidos de las pantallas y restaurando el ordenador al sistema operativo MSX.

El programa, pese a sus grandes prestaciones, es de tamaño reducido, lo que hará más fácil la introducción en nuestro ordenador, empleando un tiempo mínimo.

Al estar realizado en forma estructurada y modular, permite realizar modificaciones y ampliaciones muy fácilmente, por tanto vosotros mismos podréis añadir ciertas funciones nuevas que seguramente enriquecerán el programa.

TECLAS DE FUNCION
F1 Pasa a la siguiente pantalla (0,1,2,3)
F2 Cursor visible/invisible.
F3 Salvar en disco.
F4 Cargar en pantalla cero (pulsar D)
F5 Fin de programa

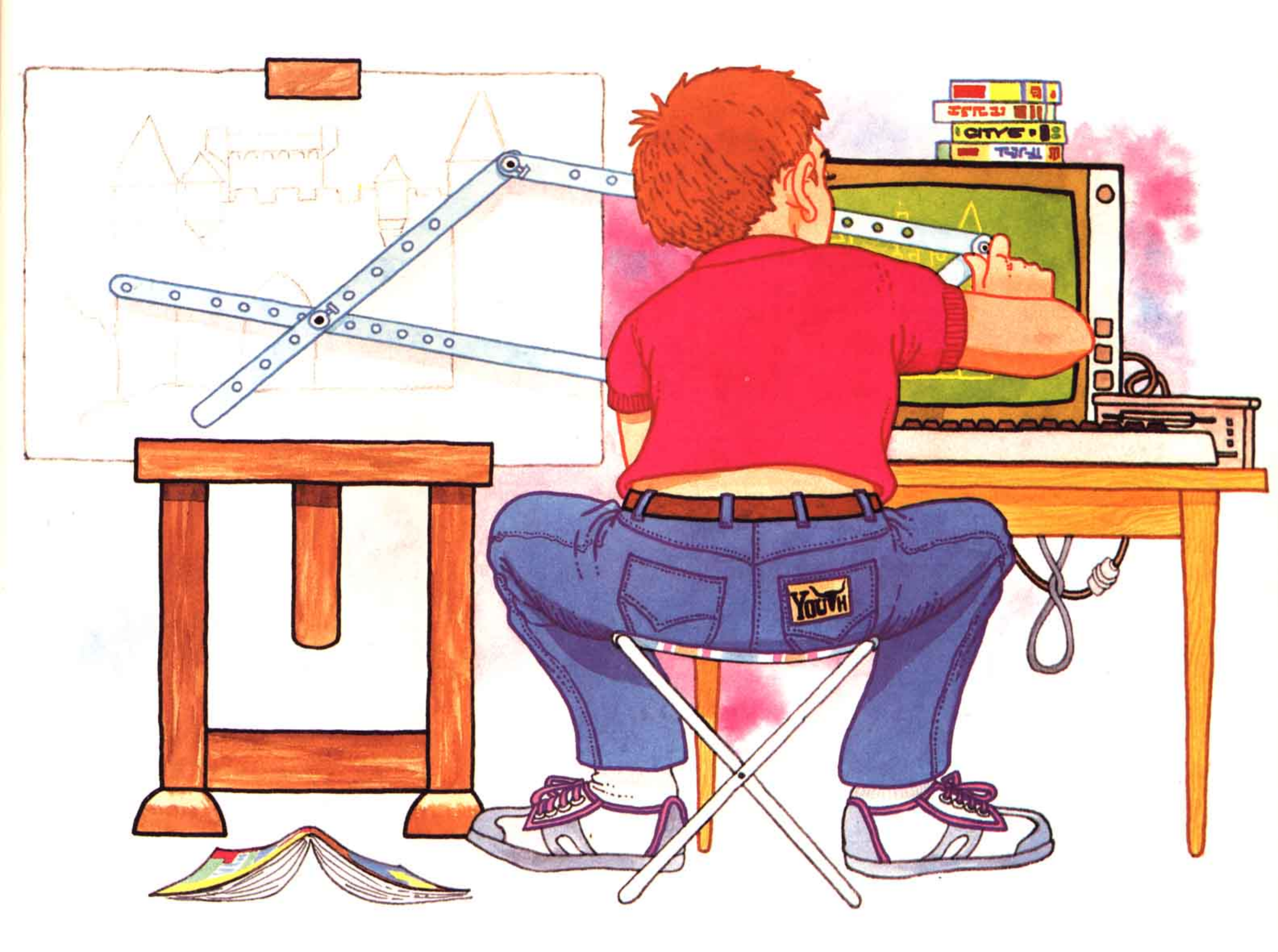
COMANDOS
^N Borra la pantalla corriente.
^C Cambia el color de la tinta.
^B Cambia el color del borde.
^P Cambia el color del papel (sin borrar).
^F Rellena hasta el límite.
^H Cambia el color del papel (borrando).
^I Fija el punto de referencia.
^L Traza una línea desde la referencia.
^Q Traza un rectángulo desde la referencia.
^O Traza un círculo con centro en la referencia.
− Selecciona paleta de colores bajos.
+ Selecciona paleta de colores altos.

RESUMEN DE LAS VARIABLES EMPLEADAS EN EL PROGRAMA.

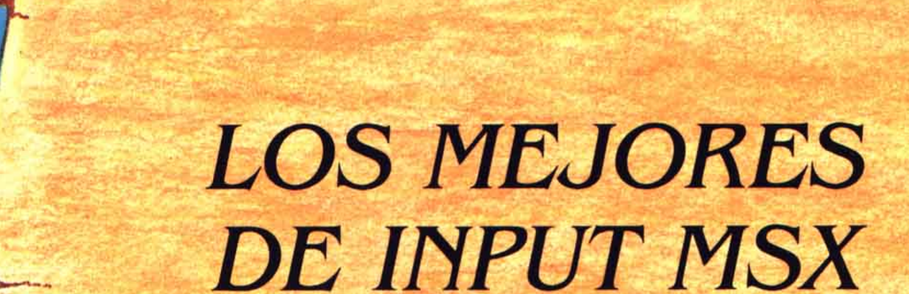
P	Pantalla corriente (0..3)
CO(4)	Color de la tinta en la pantalla.
PA(4)	Color del papel en la pantalla.
BO(4)	Color del borde en la pantalla.
SC	Paleta seleccionada (O/8).
AN	Ancho de la pantalla.
AL	Altura de la pantalla.
X,Y	Coordenadas temporales.
XN,YN	Coordenadas corrientes.
XO,YO	Coordenadas anteriores.
XI,YI	Coordenadas de referencia.
A$	Buffer de carácter.
EN	Permiso de cursor (0/1)

```
100 ' **********************
110 ' *                    *
120 ' *  Programa de dibujo
         asistido          *
130 ' *                    *
140 ' *  por ordenador.    *
150 ' *                    *
160 ' *  (msx-dos)         *
170 ' *                    *
180 ' **********************
190 '
200 ' GUIA RAPIDA DEL
    USUARIO.
210 ' ----------------------------------
220 ' F1 Siguiente pag (0,1,2,3)
230 ' F2 Punto (visible/invisible)
240 ' F3 Guardar en disco la
    pant.
250 ' F4 Cargar pant. 0 ("D")
260 ' F5 Fin de programa.
270 '
280 ' ^N Borra la pant. corriente
290 ' ^C Color de la tinta
300 ' ^B Color del borde
310 ' ^P Color del papel (lento)
320 ' ^F Rellenar contorno
330 ' ^H Color papel
    destruyendo!
340 ' ^I Fijar punto de ref.
350 ' ^L Trazar línea desde ref.
360 ' ^Q Rectángulo desde ref.
370 ' ^O Circulo desde ref.
380 '
390 ' − Col. bajos (0..7)
400 ' + Col. altos (0..7) (+8)
410 '
420 '
430 ' INICIALIZACIONES.
440 ' ----------------------------------
450 ON STOP GOSUB 2090
460 STOP ON
470 KEY OFF
480 ON KEY GOSUB840,940,
    950,1050,2140
490 KEY(1) ON:KEY(2)
    ON:KEY(3) ON
500 KEY(4) OFF:KEY(5) ON
510 SCREEN 5
520 OPEN "GRP:" AS #1
530 DIM CO(4),PA(4),BO(4)
540 FOR P=3 TO O STEP −1
550 SET PAGE P,P
560 CO(P)=3:PA(P)=6:BO(P) =
    8
570 COLOR CO(P),PA(P),BO(P)
580 CLS
590 NEXT P
600 P=0
610 EN=1
620 SC=0
630 AN=255:AL=211
640 XN=0:YN=0:XO=0:YO=0
650 CLS
660 '
670 ' BUCLE PRINCIPAL
    RAPIDO.
680 ' ----------------------------------
690 A=PAD(12)
700 XN=XO+PAD(13):YN=YO+
    PAD(14)
710 IF XN<0 THEN XN=0
720 IF XN>AN THEN XN=AN
730 IF YN<0 THEN YN=0
740 IF YN>AL THEN YN=AL
750 IF EN=1 THEN PSET (XN,
    YN)
```

```
760 IF EN=1 AND STRIG(1)=0
    THEN PRESET (XO,YO)
770 XO=XN:YO=YN
780 A$=INKEY$
790 IF A$< > "" THEN GOSUB
    1220
800 GOTO 690
810 '
820 '  (F1) SIGUIENTE PAGINA
830 '---------------------------------
840 P=(P+1)MOD 4
850 IF P=0 THEN KEY(4) ON
    ELSE KEY(4) OFF
860 PRESET (XO,YO)
870 PRESET (XN,YN)
880 SET PAGE P,P
890 COLOR CO(P),PA(P),BO(P)
900 RETURN
910 '
920 '  (F2) CURSOR (S/N)
930 '---------------------------------
940 EN=(EN+1) MOD
    2:RETURN
950 '
960 '  (F3) SALVAR EN DISCO'
970 '---------------------------------
980 A$=STR$(TIME)
990 A$=RIGHT$ (A$,LEN(A$) -
    1)
1000 IF LEN(A$) >6 THEN
     A$=RIGHT$(A$,6)
1010 IF LEN(A$) <6 THEN
     A$="0" + A$:GOTO 1010
1020 A$="DB" + A$ +".DAT"
1030 COPY (0,0) - (256,212),P
     TO A$
1040 RETURN
1050 '
1060 '  (F4) CARGAR DE DISCO
1070 '---------------------------------
1080 D$=INKEY$:IF D$=" "
     GOTO 1080
1090 IF D$< >"D" GOTO 1210
1100 SCREEN 0:CLS
1110 FILES
1120 D$=INKEY$:IF D$=" "
     GOTO 1120
1130 CLS
1140 LOCATE 0,16
1150 INPUT "Nombre completo
     del archivo a leer :"; D$
```



```
1160 SCREEN 5
1170 IF RIGHT$(D$,4) < >
     ".dat" GOTO 1200
1180 COPY D$ TO (0,0),0
1190 EN=0
1200 SET PAGE P,P
1210 RETURN
1220 '
1230 ' SERVICIO DE
     COMANDOS.
1240 ' ---------------------------------
1250 IF A$=CHR$(14) THEN
     CLS:RETURN
1260 IF A$=CHR$(3) THEN
     GOSUB 1400:RETURN
1270 IF A$=CHR$(2) THEN
     GOSUB 1490:RETURN
1280 IF A$=CHR$(16) THEN
     GOSUB 1570:RETURN
1290 IF A$=CHR$(8) THEN
     GOSUB 1710:RETURN
1300 IF A$="+" THEN
     SC=8:RETURN
1310 IF A$="–" THEN
     SC=0:RETURN
1320 IF A$=CHR$(6) THEN
     GOSUB 1800:RETURN
1330 IF A$=CHR$(9) THEN
     GOSUB 1850:RETURN
1340 IF A$=CHR$(12) THEN
     GOSUB 1900:RETURN
1350 IF A$=CHR$(17) THEN
     GOSUB 1950:RETURN
1360 IF A$=CHR$(15) THEN
     GOSUB 2030:RETURN
1370 PRINT #1,A$
1380 RETURN
1390 '
1400 ' ( ^C) COLOR DE LA
     TINTA.
1410 ' ---------------------------------
1420 A$=INKEY$:IF A$=" "
     GOTO 1420
1430 IF A$<"0" OR A$>"7"
     GOTO 1420
1440 IF PA(P)=VAL(A$)+SC
     GOTO 1470
1450 CO(P)=VAL(A$)+SC
1460 COLOR CO(P)
1470 RETURN
1480 '
1490 ' ( ^B) COLOR DEL BORDE.
1500 ' ---------------------------------
1510 A$=INKEY$:IF A$=" "
     GOTO 1510
1520 IF A$<"0" OR A$>"7"
GOTO 1510
1530 BO(P)=VAL(A$)+SC
1540 COLOR ,,BO(P)
1550 RETURN
1560 '
1570 ' ( ^P) COLOR DEL PAPEL.
1580 ' ---------------------------------
1590 A$=INKEY$:IF A$=" "
     GOTO 1590
1600 IF A$<"0" OR A$>"7"
     GOTO 1590
1610 IF CO(P)=VAL(A$)+SC
     GOTO 1700
1620 PO=PA(P)
1630 PA(P)=VAL(A$)+SC
1640 COLOR ,PA(P)
1650 FOR Y=1 TO 212
1660 FOR X=1 TO 255
1670 IF POINT (X,Y)=PO THEN
     PSET(X,Y),PA(P)
1680 NEXT X
1690 NEXT Y
1700 RETURN
1710 '
1720 ' ( ^H) COLOR PAPEL CON
     CLS.
1730 ' ---------------------------------
1740 A$=INKEY$:IF A$=" "
     GOTO 1740
1750 IF A$<"0" OR A$>"7"
     GOTO 1740
1760 PA(P)=VAL(A$)+SC
1770 COLOR,PA(P)
1780 CLS
1790 RETURN
1800 '
1810 ' ( ^F) RELLENADO DE
     COLOR.
1820 ' ---------------------------------
1830 PAINT (XN,YN), CO(P),
     CO(P)
1840 RETURN
1850 '
1860 ' ( ^I) GUARDA LA
     REFERENCIA.
1870 ' ---------------------------------
1880 XI=XN:YI=YN
1890 RETURN
1900 '
1910 ' ( ^L) LINEA DESDE REF.
1920 ' ---------------------------------
1930 LINE (XI,YI) – (XN,YN)
1940 RETURN
1950 '
1960 ' ( ^Q) RECTANGULO
     DESDE REF.
1970 ' ---------------------------------
1980 LINE (XI,YI) – (XN,YI)
1990 LINE (XI,YI) – (XI,YN)
2000 LINE (XN,YI) – (XN,YN)
2010 LINE (XI,YN) – (XN,YN)
2020 RETURN
2030 '
2040 ' ( ^0) CIRCLE CON
     CENTRO EN REF.
2050 ' ---------------------------------
2060 D=INT (SQR(((XN–XI) ^2)
     + ((YN–YI) ^2)))
2070 CIRCLE (XI,YI),D,,,,1.3
2080 RETURN
2090 '
2100 ' SERVICIO DEL STOP.
2110 ' ---------------------------------
2120 STOP ON
2130 RETURN
2140 '
2150 ' FIN DEL PROGRAMA'
2160 ' ---------------------------------
2170 KEY(1) OFF:KEY(2) OFF
2180 KEY(3) OFF:KEY(4) OFF
2190 KEY(5) OFF
2200 SCREEN 0
2210 KEY ON
2220 COLOR 15,4,4
2230 CLS
2240 CLEAR
2250 PRINT "Fin de programa."
2260 END
```



LOS MEJORES DE INPUT MSX

PUESTO	TITULO	PORCENTAJE
1.º	Soccer	23,1 %
2.º	Knight Lore	15,8 %
3.º	Hero	13,9 %
4.º	Profanation	12,1 %
5.º	Jack the Nipper	8,4 %
6.º	Alien-8	6,5 %
7.º	Riber Raid	5,5 %
8.º	Camelot Warriors	5,5 %
9.º	Yie Ar Kung-Fu II	4,6 %
10.º	Road Fighter	4,6 %
		100 %

Para la selección de esta relación únicamente se han tenido en cuenta las votaciones enviadas por nuestros lectores de acuerdo con la sección «Los Mejores de Input».

Febrero de 1987

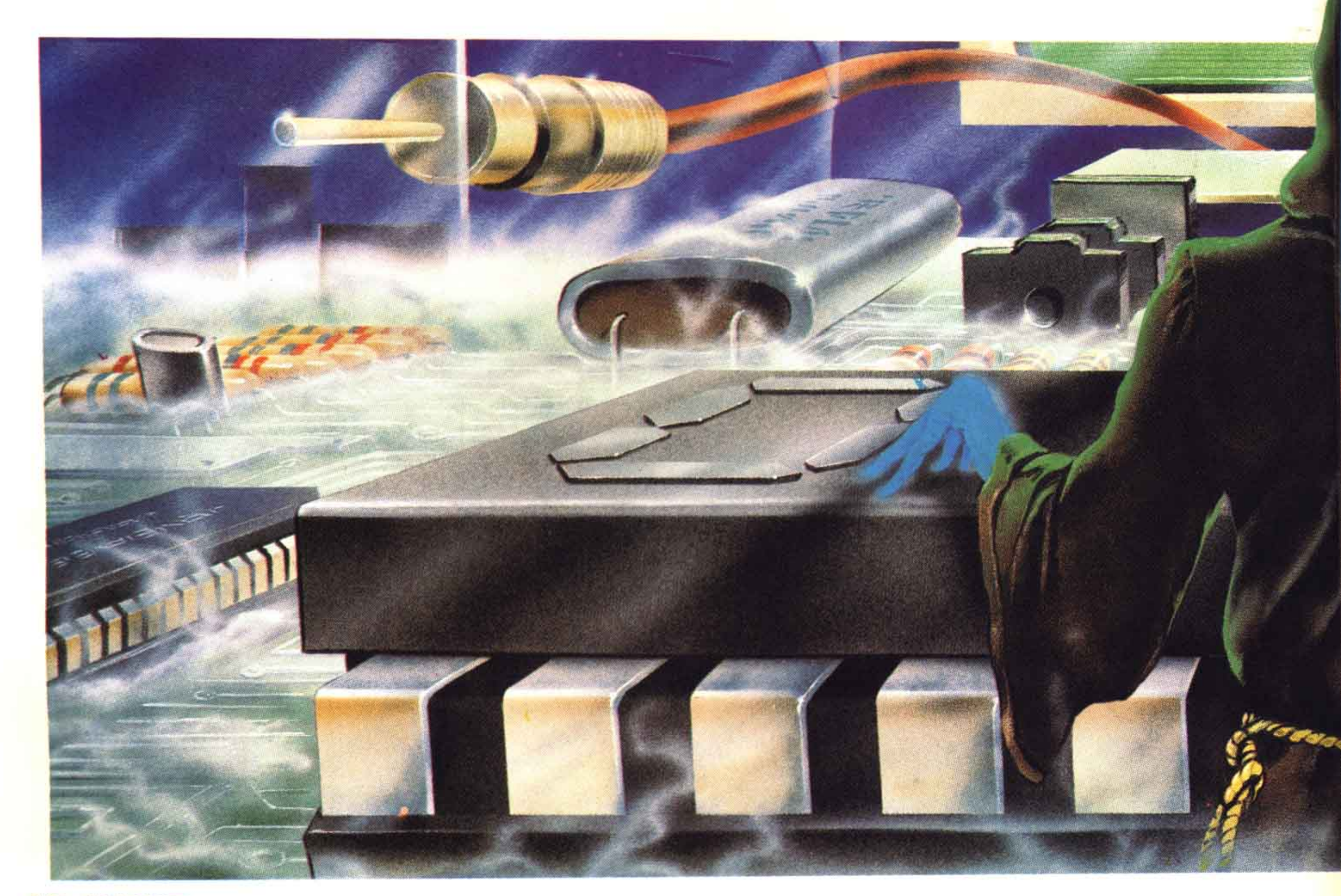


EL GRAN IMPACTO DEL MICROPROCESADOR

Nuestra revista, fiel a su espíritu de ir ampliando horizontes para sus lectores, incorpora con este número una nueva sección. En ella iniciamos la publicación de una serie de artículos, que ha sido concebida como vehículo para un mejor conocimiento de nuestro ordenador —especialmente de su parte interior—, lo que nos va a permitir sacar el mayor partido a la infinidad de posibilidades que ofrece el sistema MSX.

No por sabido está de más recordar que la creación de programas y utilidades no es tarea sencilla, pues requiere unos conocimientos que se adquieren poco a poco mediante los estudios correspondientes. Esta circunstancia no representa que para manejar un microordenador sea imprescindible saber programar. Cualquier persona puede utilizar un ordenador, en la calidad de usuario de los programas que ofrece el mercado de *software*, ciertamente muy amplio. Pero para los iniciados en el tema, y los que comienzan a sentir cierta inquietud, les es muy conveniente conocer por dentro su microordenador, analizarlo como herramienta de trabajo, aumentar el campo de conocimientos, a la par que disfruta comprobando la obtención de mayor rendimiento.

Así pues, intentaremos plasmar de forma divulgativa, lo más gráfica y sencilla posible, estas cuestiones que se crean dentro del mundo del microordenador.



Lo que sí solicitamos es perseverancia, a la ayuda que desde estas páginas os vamos a brindar para conseguir descubrir las posibilidades de que disponéis con vuestro ordenador MSX.

UN POCO DE HISTORIA

En esta primera *«inmersión»* en el mundo interior de nuestro ordenador, comenzaremos haciendo un poco de historia.

El término microordenador se ha venido empleando desde hace algunos años para denominar a todo tipo de pequeño dispositivo computador. Un microordenador no es más que un ordenador controlado por uno o más *microprocesadores*.

El microordenador ha sido el resultado de una evolución, partiendo de las familias de circuitos lógicos *TTL* (transistor transistor logic) y *MOS* (metal oxide semiconductor) de las escalas de integración media y alta (*MSI/LSI*), que son una serie de circuitos integrados que juntamente con los *reles*, fueron los protagonistas de los automatismos de otras épocas.

Este resultado evolutivo surgió ante la necesidad de simplificar el diseño de nuevas aplicaciones. Anteriormente era preciso diseñar una concepción distinta para cada aplicación concreta, que se convertía en un proceso complejo, caro y lento en algunos casos.

La llegada del *microprocesador* no hizo que se prescindiera de estos circuitos, ya que generalmente son los encargados de comunicar la *CPU (Unidad Central de Proceso)* con el resto de los circuitos electrónicos del sistema.

En la actualidad se construyen automatismos empleando aún estos circuitos, bien por ser diseños extremadamente sencillos o cuando se precisan velocidades muy altas y que un *microprocesador* no puede alcanzar, a pesar de su mayor coste.

La aparición del *microprocesador* hizo que se estandarizaran las soluciones a los problemas de diseño, ofreciendo una amplia gama de posibilidades y facilitando que con un mismo diseño estándar en *uP*, se pudiera dar respuesta a numerosos planteamientos, evitando de este modo *«las soluciones a medida»* que se utilizaban anteriormente.



Como toda nueva tecnología, también pasó por fases evolutivas. Así, al principio, los microordenadores eran muy grandes, no eran rápidos y fueron utilizados como terminales de ordenadores más potentes.

Se consideraba una tarea ardua y complicada el solo hecho de programar un ordenador, y mucho más llegar a entenderlo, mitificándolo de tal forma, que sólo estaba al alcance de un reducido grupo de personas especializadas, en un laboratorio en el que los técnicos vestían bata blanca.

Todo esto ha cambiado, desde el momento en que cualquier persona puede tener un microordenador en casa, como es el caso del MSX.

Factor determinante es que los microordenadores domésticos MSX incorporan el lenguaje de alto nivel BASIC, que permite, sin necesidad de demasiados conocimientos previos, realizar funciones básicas, abriendo así nuevos horizontes

Antiguamente, lo que a nosotros nos supone una línea de programa, significaba muchas horas de estudio para aquel señor de la bata blanca, lo que demuestra hasta qué nivel se ha simplificado la programación.

Asimismo, los circuitos clásicos han quedado relegados a un segundo término, cambiando de función. Antes, el rele servía como elemento de control, como memoria de estado de un automatismo, mientras que ahora los datos y estados internos de la máquina, se retienen en la memoria del sistema y los reles son elementos frontera. Es decir, son los componentes, que excitados por el microordenador, controlan dispositivos de potencia, dado que el *microprocesador* está limitado para trabajar con grandes potencias por sí solo.

Los primeros *microprocesadores* que se construyeron fueron de 4 *bits*, pero pronto se vio que la capacidad de memoria, datos y juego de instrucciones que podían manejar eran insuficientes.

Del *microprocesador* de 4 *bits* se pasó al de 8 *bits*, que permitía manejar cómodamente textos, ya que en cada posición de memoria se puede almacenar un carácter alfabético, así como manejar cantidades entre 0 y 255 o números con signo entre −128 y 127 en un solo acceso a memoria.

De esta serie de *microprocesadores* de 8 *bits*, el primero fue el 8008 seguido del 8080, ambos con la característica común de precisar de más de una tensión de alimentación para funcionar. Además, necesitan estar acompañados de *chips* controladores especiales que hacen posible su funcionamiento. El 8080 tiene limitada su velocidad máxima a 1Mhz. (1.000.000 de estados en cada segundo), lo cual es poco comparado con los *uP* actuales.

Estos microprocesadores fueron superados por otros como el 8085 que funciona solamente con una tensión de alimentación y posee el *bus* multiplexado, funcionando con una frecuencia de reloj máxima de 6 Mhz.

EL REVOLUCIONARIO Z80

Paralelamente al 8085 se creó el *Z80*, que absorbe y amplía las funciones del 8080 tanto en *hardware* como en *software*, pues sólo requiere una tensión de alimentación, maneja la misma cantidad de memoria, funciona a una velocidad de reloj de 4 Mhz. y es capaz de controlar memoria dinámica con una mínima electrónica externa.

Una particularidad del *Z80* es que,

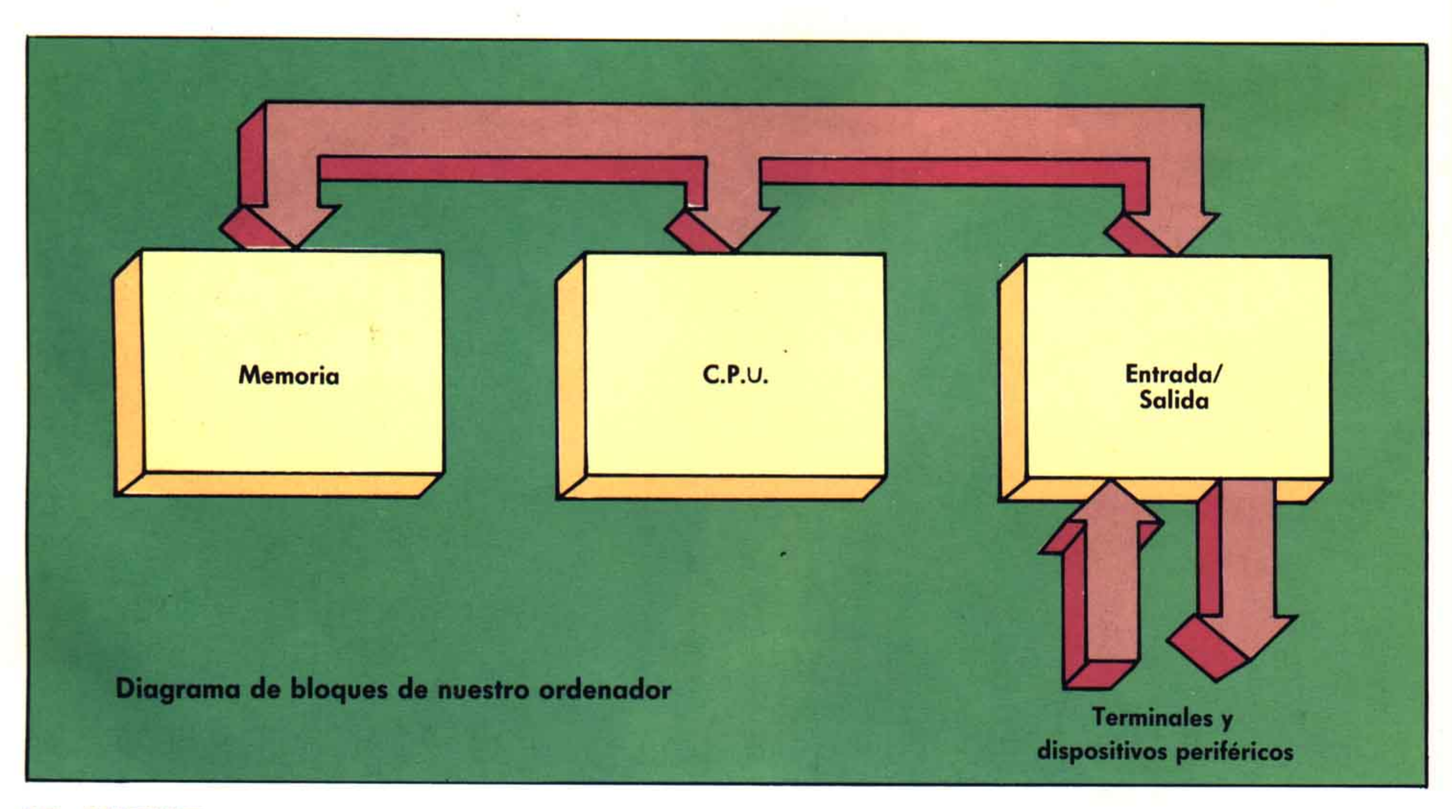
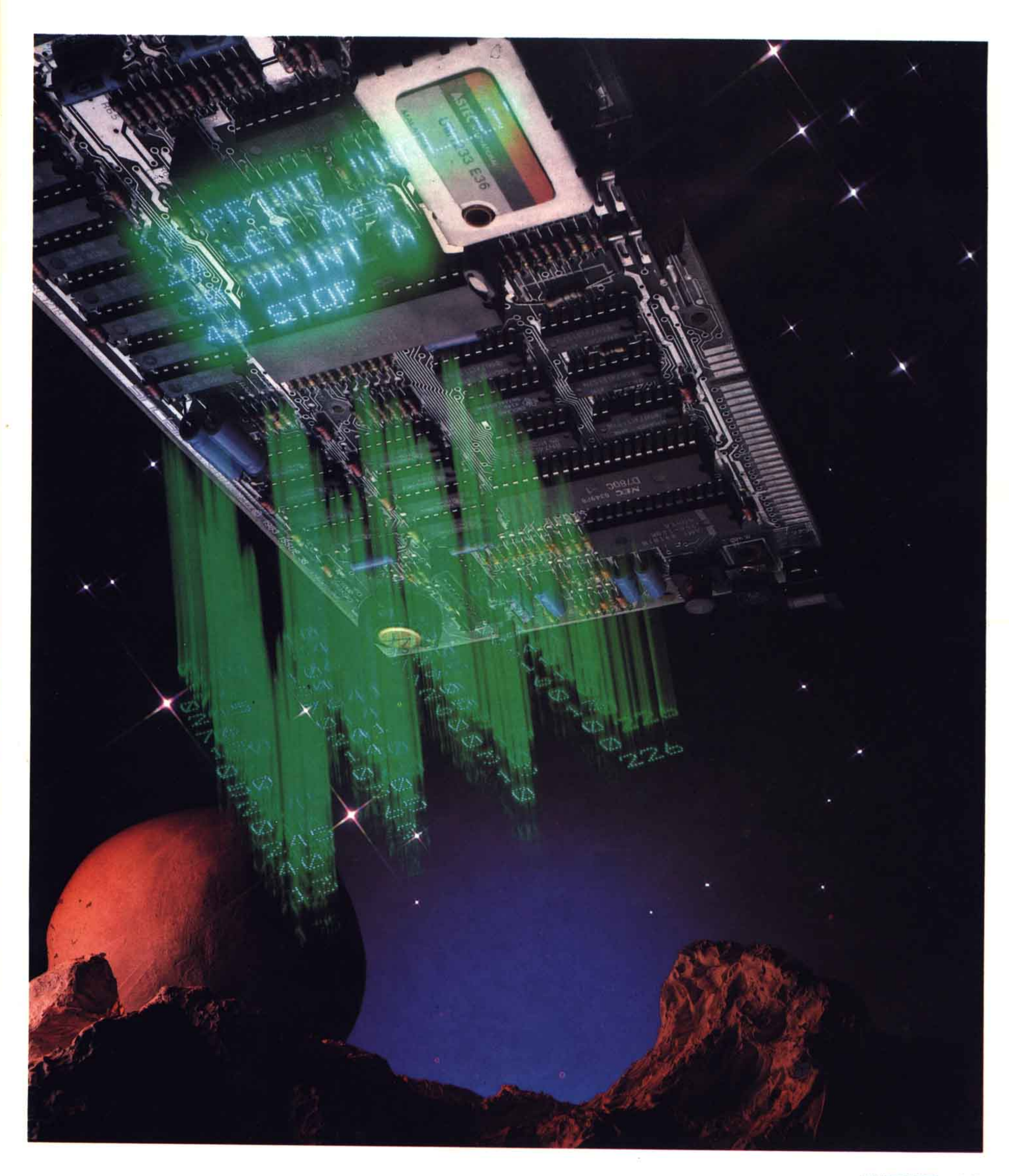


Diagrama de bloques de nuestro ordenador

El micro Z-80



como absorbe las funciones del 8080, puede sustituirlo en antiguos sistemas mejorando sus prestaciones.

La familia de componentes Z80 significó un importantísimo avance en el estado de la tecnología de los microordenadores. Estos componentes son interconectables con la mayoría de memorias estándar, generando así un amplio abanico de posibilidades de diseño de distintos microordenadores.

ESTRUCTURA DEL MICROPROCESADOR

Pero un *microprocesador* no puede realizar ninguna función por sí mismo. Necesita de una memoria que le proporcione las instrucciones a ejecutar en secuencia, así como dónde guardar los datos y resultados temporales de los cálculos realizados. Igualmente precisa de canales que hagan posible la comunicación con el exterior.

Llegamos así a la concepción de la idea de estructura del ordenador, que es en definitiva el contexto en el que se halla el *microprocesador* para constituir un sistema computador.

Esta estructura se divide en tres bloques principales que son:

—Memoria.

—CPU.

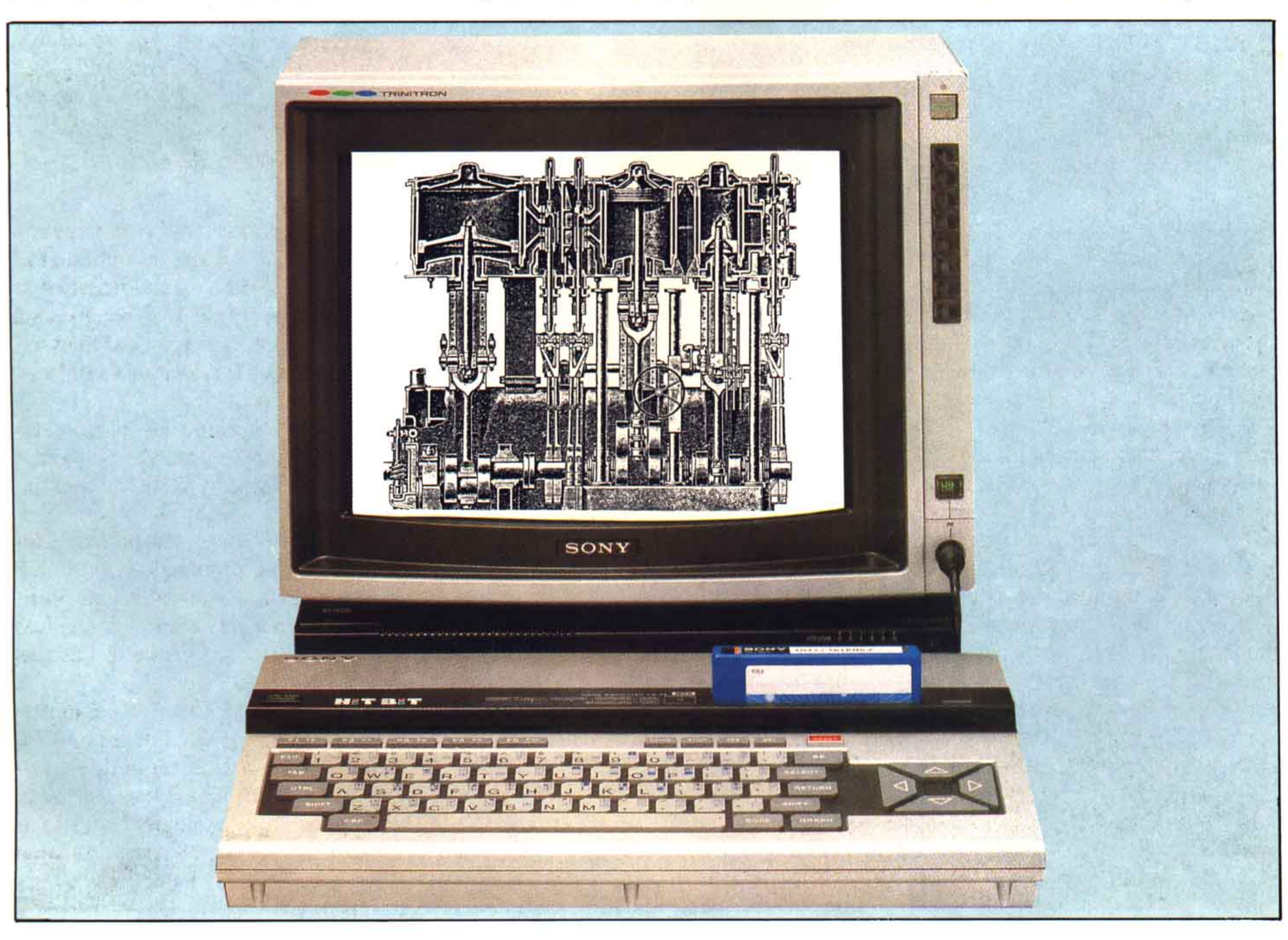
—Dispositivos de *Entrada/Salida (I/O)*.

La memoria se puede dividir en dos partes: *ROM* y *RAM*. La memoria *ROM* es inalterable y se retienen en ella los programas que hacen posible el funcionamiento de la máquina, en nuestro caso el intérprete del lenguaje BASIC y el sistema operativo básico del MSX.

La memoria *RAM*, por su parte, es donde cargaremos los programas que hemos elaborado nosotros mismos, o bien los que compremos en el comercio y también es el dispositivo donde se guardarán los datos para realizar cálculos y el resultado de los mismos. Parte de la *RAM* retiene una copia de la información mostrada por la pantalla del ordenador, ésta es la llamada *VRAM*. Debemos tener muy en cuenta que toda la información depositada en la memoria *RAM* se destruye al desconectar el ordenador.

El segundo bloque incluye la *CPU*, que es la encargada de ejecutar las instrucciones que componen los programas, y en definitiva es el componente que da *marcha* a nuestro ordenador. Contrariamente a lo que se pueda creer en un principio, la *CPU* no se detiene nunca mientras el ordenador esté en funcionamiento, ya que cuando no se ejecuta ningún programa del usuario, el control de la *CPU* es tomado por los programas que componen el sistema operativo del ordenador.

Finalmente tenemos los dispositivos





de *Entrada/Salida (Input/Output),* que tienen por misión comunicar la *CPU* con el mundo exterior. En nuestro equipo *MSX*, disponemos de ciertos circuitos especializados en determinadas funciones de entrada/salida. Por ejemplo está el *PSG (programable sound generator AY38910)* que es el encargado de generar todos los efectos de sonido. Otro ejemplo de *chip* especializado en entrada/salida, es el *VDP (video display procesor)*, que es el encargado de generar las señales de video para el monitor del sistema. Conectados a estos circuitos que genéricamente reciben el nombre de controladores, se encuentran los dispositivos periféricos. Pantallas, teclados, discos flexibles y rígidos, cassettes e impresoras, son elementos periféricos que se comunican con la memoria.

Las tres partes son imprescindibles para cualquier proceso y son las que definirán las prestaciones de nuestro equipo.

También hemos podido observar que para que un ordenador, independientemente de su complejidad, potencia, velocidad o prestaciones, realice alguna función, es imprescindible que disponga de un programa que le indique en cada momento lo que debe hacer.
Estos programas son introducidos por el usuario empleando un lenguaje de programación adecuado, (*BASIC, PASCAL, FORTRAN, ASSEMBLER* u otros), en la memoria de la máquina.

La *CPU* siempre ejecuta los programas en *«Lenguaje Máquina»*. Si el programador entra el programa en otro lenguaje, pasa por un traductor interno que lo transforma en código máquina.

De ahí que el programa que ejecutará más rápidamente el *microprocesador* será el que se le proporcione directamente en su lenguaje, debido a que ya no deberá pasar por el programa intérprete (traductor) siendo procesado por la CPU de inmediato.

Generalmente, a nivel doméstico se emplea el lenguaje *BASIC* ya que está incorporado en el propio ordenador y aporta la nada desdeñable ventaja de una fácil comprensión.

En próximos números seguiremos familiarizándonos con el microordenador, en lo que a su estructura interna se refiere. Abordaremos la arquitectura de la *CPU*, los modos de direccionamiento empleados por el *UP* para acceder a la memoria y periféricos del sistema, así como las técnicas que utiliza para comunicarse con el mundo exterior.

Asimismo, es nuestra intención describir el funcionamiento de las instrucciones del Lenguaje Máquina del *microprocesador* dado que los algoritmos codificados en este lenguaje resultan mucho más compactos y rápidos, pese a la complejidad en la confección de los mismos.

EL PALABRERO

El programa *«PALABRERO»* es una versión del conocido *«MASTER MIND»* pero trabajando con palabras en lugar de con fichas de colores.

Permite jugar a un jugador contra la máquina, o bien, a dos jugadores entre sí.

El juego consiste en adivinar la palabra de cinco letras propuesta por el otro jugador, o en su caso, por la máquina, presentando a cada intento del jugador, el número de caracteres que contiene de la palabra buscada, independientemente de su posición.

El juego está limitado en número de intentos y en tiempo, es decir, inicialmente está limitado a cuarenta intentos en la modalidad de un jugador, y de un tiempo de 45 segundos por intento.

Si el jugador desea rendirse, puede pulsar la tecla "F1" y entonces el ordenador le mostrará la palabra que retenía en su interior.

Este juego puede ser considerado didáctico, dado que requiere pensar palabras lógicas que contengan algunas letras de combinaciones anteriores para llegar a la solución.

En la modalidad de un jugador, el programa tiene en su interior cien palabras para adivinar, número suficientemente elevado como para garantizar que la palabra escogida por la máquina, nos va a dar que pensar...

Advertencia

Cuando se haga correr el programa, las palabras deberán ser introducidas en letras mayúsculas, dado que tal como está diseñado, no interpreta los caracteres en minúsculas.

Partes del programa

Para los interesados en analizar la composición del programa, enumeramos a continuación los bloques componentes del mismo:

100- 240	presentación
250- 260	programación de F1
280-	inicialización de variables A y B
290- 330	inicia el sprite X$ (movimiento)
340- 360	movimiento de sprite, salida a 470
370-	música inicial
390-	bucle de espera
440-	apaga el sprite
460-	repite desde 290
470-	abre pantalla gráfica para texto
480- 540	estrellas
550- 620	dibuja el trapecio y lo rellena
630-1390	continúa la presentación puntos y circ
1400-1470	fin de la presentación
1480-1630	juego (inicio núm. jug)
1640-	juego si dos jugadores
2730-	juego si un jugador
3670-3710	palabras internas
3720-3760	escribe la solución
3770-3810	advertencia de limitaciones
3820-3890	emite un sonido

COMENTARIOS

El juego resulta divertido, ya que el conjunto interno de palabras es muy amplio.

Los límites tanto en tiempo para dar la respuesta como en número de intentos, son correctos y aumentan la emoción del juego.

Respecto a la presentación del juego, debemos decir que es muy bonita y *"galáctica"*, pero nuestro consejo en este caso es el meditar un poco sobre la longitud y eficiencia de los programas.

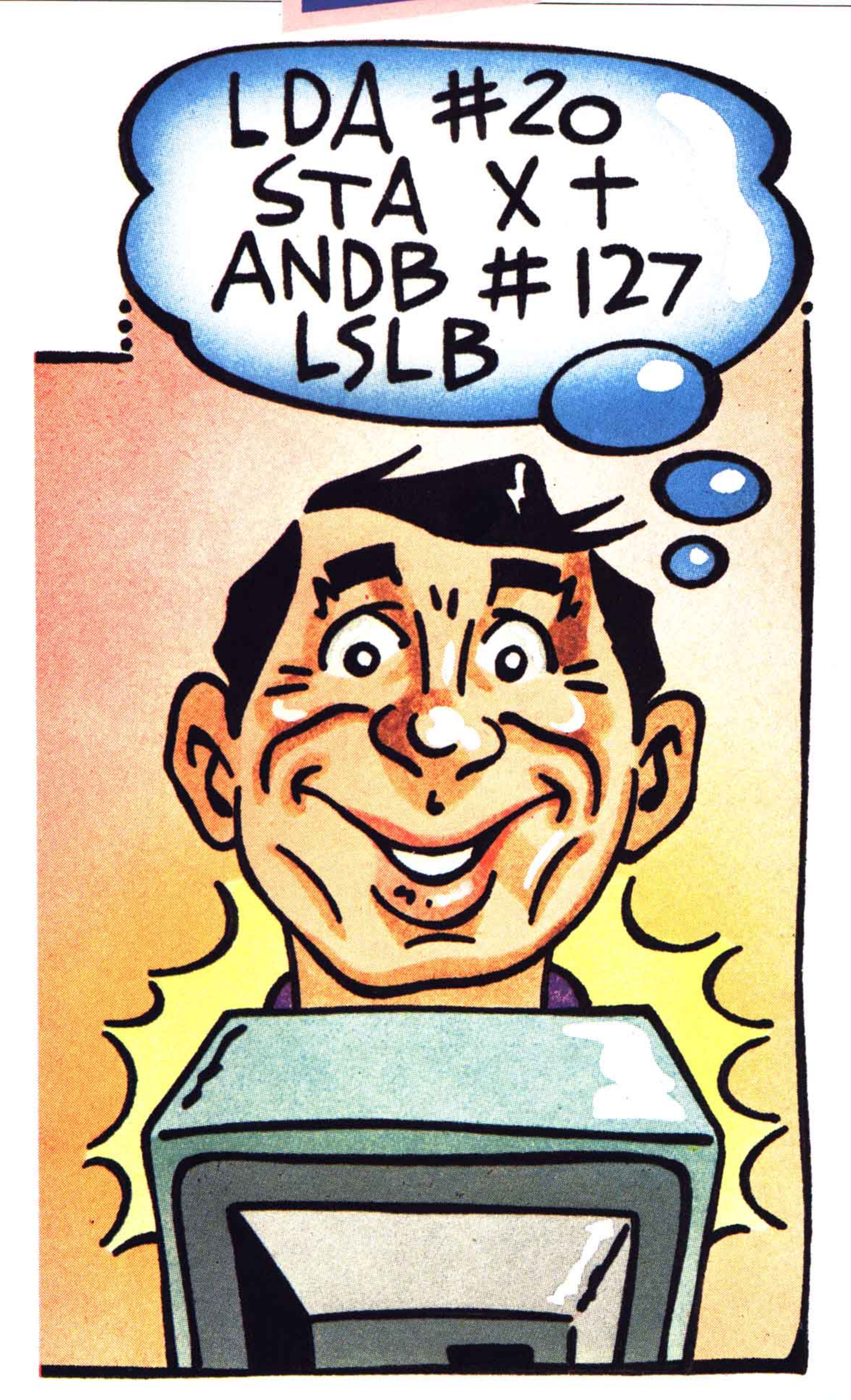
Un programa no es mejor por ocupar más espacio sino todo lo contrario; si en dos programas que poseen las mismas prestaciones, uno de ellos es más reducido, tendrá las siguientes ventajas:

—Al transcribirlo de la revista al ordenador, tendremos menos posibilidades de cometer errores.

—El tiempo empleado en este trabajo será menor.

—Ocuparemos menos memoria de la máquina.

Debido a esto, aconsejamos estructurar un poco los programas, con lo que aumentará la eficiencia


LDA #20
STA X+
ANDB #127
LSLB

de tus horas como programador.

La idea es buena, y el programa funciona.

```
1ØØ ' ===============
11Ø '
12Ø '   MASTER MIND -
    PALABRERO
13Ø '
14Ø ' ===============
15Ø '
16Ø ' *********************
17Ø ' * Programado por:     *
18Ø ' *                     *
19Ø ' *   LUIS GARCIA
    RODRIGUEZ               *
2ØØ ' *                     *
21Ø ' * (SALAMANCA. JUNIO DE
    1986)                   *
22Ø ' *********************
23Ø '
24Ø '
25Ø ON KEY GOSUB 372Ø
26Ø KEY (1) ON
27Ø COLOR 8.1,1:SCREEN 2,2
28Ø A=1:B=8
29Ø FOR I%=1TØ32
3ØØ READC%
31Ø X$=X$+CHR$(C%)
32Ø NEXT I%
33Ø SPRITE$ (1) = X$
34Ø D = D + 8 : E = 16Ø
35Ø IF D>23Ø THEN 47Ø
36Ø PUTSPRITE1,(D,E),A,
    1:BEEP
37Ø PLAY"L2ADEFFDEAADEFF
    EDAGGG"
38Ø SWAP A, B
39Ø FOR J = 1TO 1Ø:NEXTJ
4ØØ DATA Ø,111, 111, 1Ø8,
    1Ø8, 1Ø8, 1Ø9, 1Ø9
41Ø DATA 1Ø8, 1Ø8, 111, 111,
    96, 127, 127, Ø
42Ø DATA Ø,188, 19Ø, 54, 54,
    62, 188, 184
43Ø DATA 188, 182, 182, 182,
    Ø, 254, 254, Ø
44Ø SPRITE OFF
45Ø RESTORE: X$=""
46Ø GOTO 29Ø
47Ø OPEN "GRP: "AS #1
48Ø X = 1
49Ø H = RND(-TIME)
5ØØ FOR W = 1 TO 26Ø
51Ø X=X+1 : IF X>15 THEN X
    = 2
52Ø K = INT (RND (1) * 192)
53Ø PSET (W, K), X
54Ø NEXT
55Ø COLOR 4
56Ø PSET (8,136)
57Ø LINE -(56,88)
58Ø PSET (248,136)
59Ø LINE -(2ØØ,88):LINE -
    (56,88)
6ØØ PSET (8,136)
61Ø LINE -(248,136)
62Ø PAINT(58,9Ø)
63Ø LINE (8,136) - (248,144),
    1Ø, B
64Ø PAINT (1Ø,138), 1Ø
65Ø COLOR 13
66Ø PSET (48,12Ø)
67Ø LINE -(64,48)
68Ø PSET (64,48)
69Ø LINE -(76,48) : LINE -
    (8Ø,6Ø) : LINE - (84,48) :
    LINE - (96,48) : LINE -
    (112,12Ø) : LINE - (96,
    12Ø) : LINE - (92,1ØØ) :
    LINE - (88,12Ø)
7ØØ PSET (48,12Ø)
71Ø LINE -(64,12Ø) : LINE -
    (68,1ØØ) : LINE - (72,12Ø)
    : LINE - (88,12Ø)
72Ø PAINT (9Ø,5Ø)
73Ø PSET (212,12Ø)
74Ø LINE -(196,12Ø)
75Ø PSET (2Ø4,84)
76Ø LINE -(212,12Ø)
77Ø PSET (2Ø4,84)
78Ø LINE -(212,48) : LINE -
    (196,48) : LINE - (192,68)
    : LINE - (188,48) : LINE -
    (172,48) : LINE - (18Ø,84)
    : LINE - (172,12Ø) : LINE -
    (188,12Ø) : LINE - (192,
    1ØØ) : LINE - (196,12Ø)
79Ø PSET (168,64)
8ØØ LINE -(168,48)
81Ø PSET (116,1Ø4)
82Ø LINE -(116,12Ø)
83Ø LINE -(164,12Ø)
84Ø PSET (116,1Ø4)
85Ø LINE -(148,1Ø4)
86Ø PSET (168,48)
87Ø LINE -(12Ø,48)
88Ø PSET (168,64)
89Ø LINE -(136,64)
9ØØ CIRCLE (152,96), 8,,
    4.24873, .
    4636476Ø9ØØ81#
91Ø CIRCLE (132,72), 8,,
    1.1Ø71487177941 #,
    3.6Ø5237#
92Ø CIRCLE (136,72), 28,,
    2.158797#, 3.926988#
93Ø CIRCLE (148,1ØØ), 25,,
    5.387125#,
    1.249Ø457723982#
94Ø PSET (156,76)
95Ø LINE -(124,76)
96Ø LINE -(124,72)
97Ø PSET (116,92)
98Ø LINE -(16Ø,92)
99Ø LINE -(16Ø,96)
1ØØØ PSET (116,91)
1Ø1Ø PSET (149,1Ø4)
1Ø2Ø PSET (135,64)
1Ø3Ø PAINT (136,49)
1Ø4Ø PAINT (177,49)
1Ø5Ø PSET (14Ø,Ø)
1Ø6Ø LINE -(64,48)
1Ø7Ø PSET (76,48)
1Ø8Ø LINE -(14Ø,Ø)
1Ø9Ø LINE -(84,48)
11ØØ PSET (96,48)
111Ø LINE -(14Ø,Ø)
112Ø LINE -(112,56)
113Ø PSET (168,48)
114Ø LINE -(14Ø,Ø)
115Ø LINE -(172,48)
116Ø PSET (188,48)
117Ø LINE -(14Ø,Ø)
118Ø LINE -(196,48)
119Ø PSET (212,48)
12ØØ LINE -(14Ø,Ø)
121Ø PSET (172,92)
122Ø LINE -(164,64)
123Ø PSET (116,1Ø4)
124Ø LINE -(12Ø,92)
125Ø PSET (112,12Ø)
126Ø LINE -(116,1Ø4)
127Ø PSET (64,121)
128Ø LINE -(7Ø,113)
129Ø PSET (89,12Ø)
```

```
13ØØ LINE -(94,113)
131Ø PSET (171,12Ø)
132Ø LINE -(169,114)
133Ø PSET (195,12Ø)
134Ø LINE -(19Ø,112)
135Ø PSET (179,84)
136Ø LINE -(168,65)
137Ø COLOR 15
138Ø LINE(169,155)-(9Ø,177),,
     B
139Ø LINE (86,181)-(173,
     151),,B
14ØØ A=1:B=15
141Ø FOR X=
     Ø TO 6ØØ STEP 1Ø
142Ø SWAP A,B
143Ø COLOR A
144Ø PRESET (96,163):PRINT
     #1, "PALABRERO"
145Ø NEXT X
146Ø CLOSE#1
147Ø PU=4Ø
148Ø SCREEN2:COLOR 15,4,4:
     CLS:KEY OFF:OPEN"GRP:
     "AS#1
149Ø ON KEY GOSUB 372Ø:
     KEY(1)ON
15ØØ D=Ø:Y=4:F=22:L=2:K=
     2
151Ø LINE(183,4Ø)-(7Ø,16),
     15,BF
152Ø LINE(185,42)-(68,14),
     15,B
153Ø COLOR1:PRESET(8Ø,25):
     PRINT#1,"EL
     PALABRERO"
154Ø LINE(8,64)-(248,1ØØ),1,
     BF
155Ø COLOR15:PRESET(24,8Ø):
     PRINT#1,"Por: LUIS
     GARCIA RODRIGUEZ"
156Ø COLOR 1Ø:PRESET(Ø,
     15Ø):PRINT#1,"-------
     -------"
157Ø COLOR 1:PRESET(27,
     165):PRINT#1,
     "CUANTOS JUGADORES:
     1 - 2?"
158Ø COLOR 1Ø:PRESET(Ø,
     18Ø):PRINT#1,"_______
     ________"
159Ø CLOSE#1
16ØØ K$=INKEY$
161Ø IFK$="1"THEN 273Ø
162Ø IFK$="2"THEN 164Ø
163Ø IFK$<>"2"THEN 16ØØ
164Ø SCREENØ:CLS:LOCATE12,
     Ø:PRINT"INSTRUCCIONE
     S":LOCATE12,1:PRINT"-
     ---------"
165Ø PRINTTAB(5) "En esta
     opción se trata de
     introducir, cuando el
     ordenador lo pida, una
     palabra cada jugador"
166Ø PRINTTAB(5) "A
     continuación aparecerán en
     pantalla los nombres de los
     contendientes, un contador
     de tiempo entre ambos y
     unas flechas junto al
     nombre del jugador al que
     se trata de adivinar la
     palabra."
167Ø PRINTTAB(5) "Las palabras
     se van introduciendo antes
     de que el contador llegue a
     45 segundos, de lo
     contrario el turno pasará al
     siguiente jugador."
168Ø PRINTTAB(5) "Como es
     lógico, ya que se trata de
     descubrir la palabra del
     contrario debemos
     escribirlas debajo del
     nombre de éste."
169Ø LOCATEØ,22:PRINT
     "PULSA UNA TECLA...."
17ØØ A$=INKEY$:IFA$= "
     "THEN17ØØ
171Ø CLS:PRINTTAB(5) "Al lado
     de cada palabra ingresada
     el ordenador dirá cuántas
     letras de ésta, están
     contenidas en la del
     adversario."
172Ø LOCATEØ,22:
     PRINT"PULSA UNA
     TECLA..."
173Ø A$=INKEY$:IFA$="
     "THEN173Ø
174Ø GOSUB 377Ø
175Ø CLS:Z$ = "NOMBRE DEL
     JUGADOR NUM. 1"
176Ø PRINTZ$;:INPUT M$
177Ø CLS:PRINT"NOMBRE DEL
     JUGADOR NUM. 2";
178Ø INPUT N$
179Ø CLS:LOCATE Ø,
     22:PRINT"CUANDO ESTE
     PULSA << RETURN >> "
18ØØ LOCATEØ,Ø:PRINTM$
     "ESCRIBE TU PALABRA Y
     PROCURA":PRINT"QUE
      NOLA VEA";N$:PRINT:
     PRINT;:INPUT Y$
181Ø GOSUB 261Ø
182Ø CLS:LOCATE Ø,22:PRINT
     "CUANDO ESTE PULSA<<
     RETURN >> "
183Ø LOCATEØ,Ø:PRINTN$"
     ESCRIBETUPALABRAY
     PROCURA":PRINT"QUENO
     LAVEA";M$:PRINT:PRINT;
     :INPUT W$
184Ø GOSUB 267Ø
185Ø CLS:FOR L=4TO23:
     LOCATE17,L:PRINT"V":
     NEXTL
186Ø COLOR 2,1:LOCATEY+1,Ø
     :PRINTM$
187Ø LOCATE Y,1:PRINT"____
     _________"
188Ø LOCATE 22,Ø:PRINTN$
189Ø LOCATE21,1:PRINT"____
     _____"
19ØØ GOTO 192Ø
191Ø LOCATEY+7,K:PRINT"-";
     A:LOCATE Ø,Ø:PRINT"
     ":LOCATE 35,Ø:
     PRINT"<<":GOTO 215Ø
192Ø SOUND Ø,Ø:GOSUB 261Ø
     :LOCATEØ,Ø:PRINT">>":
     LOCATE35,Ø:PRINT"  "
193Ø TIME=Ø
194Ø MIN%=TIME/25ØØ
195Ø SEG%=TIME/5Ø-
     MIN%*5Ø
196Ø LOCATE 16,
     1:PRINTSEG%
197Ø IF SEG%<35 THEN
     GOSUB 382Ø
198Ø IF SEG%=45THEN 215Ø
199Ø K$=INKEY$:IFK$="
     "THEN 194Ø
2ØØØ SOUND Ø,Ø
```

```
2Ø1Ø X=Ø:A=Ø:K=K+1:IF
     K=>22 THEN K=3
2Ø2Ø LOCATEY+2,
     K:PRINT"       ":
     LOCATE Y+2,K:PRINTK$;
     :LINE INPUT R$:LOCATE
      15,K:PRINT" V":
     R$=LEFT$(R$,4):
     X$=K$+R$
2Ø3Ø C=C+1:LOCATE Ø,1:
     PRINT C
2Ø4Ø IF X$=Y$ THEN 237Ø
2Ø5Ø X=X+1
2Ø6Ø IF X>5 THEN 191Ø
2Ø7Ø R$=MID$(X$,X,1)
2Ø8Ø IFR$=A$THEN A$="":A
     =A+1:GOTO 2Ø5Ø
2Ø9Ø IFR$=B$THEN B$="":A
     =A+1:GOTO 2Ø5Ø
21ØØ IFR$=C$THEN C$="":A
     =A+1:GOTO 2Ø5Ø
211Ø IFR$=D$THEN D$="":A
     =A+1:GOTO 2Ø5Ø
212Ø IFR$=E$THEN E$="":A
     =A+1:GOTO 2Ø5Ø
213Ø GOTO 2Ø5Ø
214Ø LOCATE F+7,L:PRINT"
     –";B:LOCATE 35,Ø:
     PRINT"  ":LOCATEØ,
     Ø:PRINT">>":GOTO192Ø
215Ø SOUND Ø,Ø:GOSUB 267Ø
     :LOCATEØ,Ø:PRINT"  ":
     LOCATE35,Ø:PRINT"<<"
216Ø TIME=Ø
217Ø MIN%=TIME/25ØØ
218Ø SEG%=TIME/5Ø–
     MIN%*5Ø
219Ø LOCATE 16,1:
     PRINTSEG%
22ØØ IF SEG%>35 THEN
     GOSUB 382Ø
221Ø IF SEG%=45THEN 192Ø
222Ø K$=INKEY$:IFK$="
     "THEN 217Ø
223Ø SOUND Ø,Ø
224Ø X=Ø:B=Ø:L=L+1:IF L=
     >22 THEN L=3
225Ø LOCATE F+2,
     L:PRINT"       ":LOCATE
     F+2,L:PRINTK$;:LINE
     INPUT B$:T$=K$+B$
226Ø IF T$=W$ THEN N$=
     M$:GOTO 237Ø
227Ø D=D+1:LOCATE 32,1:
     PRINT D
228Ø X=X+1
229Ø IF X>5 THEN 214Ø
23ØØ R$=MID$(T$,X,1)
231Ø IFR$=F$THENF$="":B=
     B+1:GOTO 228Ø
232Ø IFR$=G$THENG$="":B=
     B+1:GOTO 228Ø
233Ø IFR$=H$THENH$="":B=
     B+1:GOTO 228Ø
234Ø IFR$=I$THENI$="":B=B
     +1:GOTO 228Ø
235Ø IFR$=J$THENJ$="":B=B
     +1:GOTO 228Ø
236Ø GOTO 228Ø
237Ø CLS:P$="ESA ES LA
     PALABRA":Ø$ =
     "ENHORABUENA":I$ =
     "OTRA PARTIDA? (S/N)"
238Ø COLOR 13:D=D+1:L=L
     +3:T=4:PRINT:PRINT
2385 LOCATE T,L:PRINT
      SPACE$(38)
239Ø FOR W=1 TO LEN(P$)
24ØØ LOCATEF+1Ø,1:PRINTD
241Ø LOCATE T,L:PRINT
      MID$(P$,1,W);
242Ø R=TIME
243Ø IF TIME<R+5 THEN
     243Ø
244Ø NEXT W
245Ø L=L+3:PRINT:PRINT
2455 LOCATE T,L:PRINT
      SPACE$(38)
246Ø FOR W=1 TO LEN(Ø$)
247Ø LOCATEF+1Ø,1:PRINT D
248Ø LOCATET+5,L:PRINT
      MID$(Ø$,1,W);
249Ø R=TIME
25ØØ IF TIME<R+5 THEN
      25ØØ
251Ø NEXT W
252Ø LOCATE T+18,L:PRINTN$
253Ø L=L+3:PRINT:PRINT
2535 LOCATE T,L:PRINT
     SPACE$(38)
254Ø FOR W=1 TO LEN(I$)
255Ø LOCATEF+1Ø,1:PRINT D
256Ø LOCATET+1Ø,L:PRINT
      MID$(I$,1,W);
257Ø R=TIME
258Ø IF TIME>R+5 THEN
     258Ø
259Ø NEXT W
26ØØ GOTO 348Ø
261Ø A$=MID$(Y$,1,1)
262Ø B$=MID$(Y$,2,1)
263Ø C$=MID$(Y$,3,1)
264Ø D$=MID$(Y$,4,1)
265Ø E$=MID$(Y$,5,1)
266Ø RETURN
267Ø F$=MID$(W$,1,1)
268Ø G$=MID$(W$,2,1)
269Ø H$=MID$(W$,3,1)
27ØØ I$=MID$(W$,4,1)
271Ø J$=MID$(W$,5,1)
272Ø RETURN
273Ø '
274Ø 'SOLO UN JUGADOR
275Ø '
276Ø SCREENØ:CLS:K$="
     ":LOCATE12,Ø:PRINT"
     INSTRUCCIONES
     ":LOCATE12,1:PRINT"---
     ---------"
277Ø PRINTTAB(5) "El
     ordenador va a elegir una
     palabra que debes
     descubrir por deducción,
     ingresando sucesivamente
     palabras de 5 letras.»
278Ø PRINT
279Ø PRINTTAB(5) "Tras de ello
     el ordenador te irá diciendo
     el núm. de letras
     contenidas en la palabra a
     descubrir."
28ØØ PRINT
281Ø PRINTTAB(5) "Para ello
     dispones de 4Ø
     oportunidades y de 45
     segundos para cada una de
     ellas. Si sobrepasas alguna
     de las dos el ordenador te
     dirá cuál era la palabra."
282Ø PRINT
283Ø PRINTTAB(5) "Si en
     cualquier momento de la
     partida te das por vencido
     sabrás la palabra pulsando
     << F1 >> ."
284Ø LOCATEØ,
```




```
      22:PRINT"PULSA UNA
      TECLA..."
2850  A$=INKEY$:IFA$="
      "THEN 2850
2860  GOSUB 3770
2870  CLS:Y=4
2880  Z$="CUAL ES TU
      NOMBRE"
2890  PRINTZ$;:INPUT M$
2900  CLS:PRINT M$" ESPERA
      UN MOMENTO POR
      FAVOR"
2910  FOR S=1 TO 1500:NEXT S
2920  GOSUB 3580
2930  CLS:COLOR 2,
      1:LOCATEY+1,
      Ø:PRINTM$
2940  LOCATEY, 1:PRINT"____
      __________"
2950  K=2
2960  GOTO 3000
2970  K=K+1:PU=PU-
      1:LOCATE Y+7,
      K:PRINT"-"; A
2980  C=C+1:LOCATEØ,
      1:PRINTC
2990  IF C=40 THEN 3720
3000  TIME=Ø
3010  MIN%=TIME/2500
3020  SEG%=TIME/50-
      MIN%*50
3030  LOCATE 33,Ø:PRINTSEG%
3040  IF SEG%>35 THEN
       GOSUB 3820
3050  IF SEG%=45 THEN3720
3060  K$=INKEY$:IF K$="
      "THEN 3010
3070  SOUND Ø,Ø
3080  GOSUB 3520
3090  X=Ø:A=Ø
3100  IF K=20 THEN 3110
      ELSE 3130
3110  K=Ø:FOR L=ØTO22:
      LOCATE19,L:PRINT
      "V":NEXTL
3120  Y=22
3130  LOCATEY+2,
      K+1:PRINT"
      ":LOCATEY+2,K+1:
      PRINTK$;:LINE INPUT
      LI$:X$=K$+LI$
3140  IF X$=Q$ THEN 3240
3150  X=X+1
3160  IF X>5 THEN GOSUB
      3520:GOTO 2970
3170  R$=MID$(X$,X,1)
3180  IFR$=A$THEN A$="
      ":A=A+1:GOTO 3150
3190  IFR$=B$THEN B$="
      ":A=A+1:GOTO 3150
3200  IFR$=C$THEN C$="
      ":A=A+1:GOTO 3150
3210  IFR$=D$THEN D$="
      ":A=A+1:GOTO 3150
3220  IFR$=E$THEN E$="
      ":A=A+1:GOTO 3150
3230  GOTO 3150
3240  CLS:Y=4:P$="ESA ES LA
      PALABRA":Ø$=
      "ENHORABUENA":I$=
      "OTRA   PARTIDA? (S/N)"
3250  COLOR 13:C=C+1:K=
      K+3:T=4:PRINT:PRINT
3255  LOCATE T,K:PRINT
      SPACE$(38)
3260  FOR W=1 TO LEN(P$)
3270  LOCATEY-3,1:PRINTC
3280  LOCATEY,K:PRINT
      MID$(P$,1,W);
3290  R=TIME
3300  IF TIME<R+5 THEN 3300
3310  NEXT W
3320  K=K+3:PRINT:PRINT:
      PRINT"
```

```
3325 LOCATE T,K:PRINT
      SPACE$(38)
333Ø FOR W=1 TO LEN(Ø$)
334Ø LOCATEY–3,1:PRINTC
335Ø LOCATEY+5,K:PRINT
      MID$(Ø$,1,W);
336Ø R=TIME
337Ø IF TIME<R+5 THEN
      337Ø
338Ø NEXT W
339Ø LOCATE Y+18,
     K:PRINTM$
34ØØ K=K+3:PRINT:
     PRINT"        "
34Ø5 LOCATE T,K:PRINT
      SPACE$(38)
341Ø FOR W=1 TO LEN(I$)
342Ø LOCATEY–3,1:PRINTC
343Ø LOCATEY+1Ø,K:PRINT
     MID$(I$,1,W);
344Ø R=TIME
345Ø IF TIME<R+5 THEN 345Ø
346Ø NEXT W
3465 PRINT:PRINT
347Ø LOCATE25,K+3:PRINT
     "PUNTOS–";PU
348Ø J$=INKEY$:IF J$="
     "THEN 348Ø
349Ø IF J$="S"OR J$="s"THEN
     RUN 147Ø
35ØØ IF J$="N"OR
     J$="n"THEN NEW
351Ø IFJ$<>"S"ORJ$<
     >"N"ORJ$<>"s"ORJ$<
     >"n"THEN 348Ø
352Ø A$=MID$(Q$,1,1)
353Ø B$=MID$(Q$,2,1)
354Ø C$=MID$(Q$,3,1)
355Ø D$=MID$(Q$,4,1)
356Ø E$=MID$(Q$,5,1)
357Ø RETURN
358Ø DIM DA$(1ØØ)
359Ø RESTORE 367Ø
36ØØ FOR J=1 TO 1ØØ
361Ø READ DA$(J)
362Ø NEXT J
363Ø R=RND(–TIME)
364Ø R=INT(RND(1)*1ØØ)+1
365Ø Q$=DA$(R)
366Ø RETURN
367Ø DATA OZONO,SANTA,
     GRIFO,MUECA,TRONO,
     MARCA,FRESA,METRO,
     AZADA,GAFAS,ARBOL,
     AZOTE,TRAMA,CENIA,
     TRAPO,MUCHO,MUARE,
     RESMA,TRACA,ZUECO
368Ø DATA AGRAZ,BASTO,
     BINGO,RELOJ,MALLA,
     TRIGO,MUSGO,PAPEL,
     MIRLO,TINTA,LUNAR,
     MARTE,TROVA,CHEPA,
     PLUMA,LAPIZ,CORTA,
     LARGA,MECHA,PERLA
369Ø DATA VELLO,POEMA,
     OREJA,BOTON,GRUTA,
     VERDE,HUSMO,HURON,
     RISCO,PRADO,MULEO,
     GENTE,MUJER,TABLA,
     PARIA,PEANA,ZORRA,
     YESCA,VICIO,UJIER
37ØØ DATA UNETA,TRIPA,
     RASPA,SENIL,LUCIO,
     ARANA,GARZA,OROYA,
     PECIO,ICONO,CURVA,
     CUSCA,IDOLO,FIRMA,
     TOSCO,MUGRE,DISCO,
     SARAO,ESCOA,GAMBA
371Ø DATA FATUO,INOPE,
     LACHA,NEUMA,INANE,
     JERGA,OLIVO,HACHA,
     NIVEO,REGIO,POLEO,
     PISTA,OMASO,MOCHA,
     OCELO,ORGIA,GORRO,
     CAUCA,LINEA,ISTMO
372Ø SOUNDØ,Ø:LOCATE 8,K+
     2:PRINT"LA PALABRA
     ERA ";Q$:LOCATE23,K+3
     :PRINT"ÃÃÃÃÃÃÃÃÃÃ"
373Ø LOCATE 1,K+5:PRINT
     "PARA OTRA PARTIDA
     PULSA UNA TECLA"
374Ø R$=INKEY$:IFR$="
     "THEN 374Ø
375Ø RUN 148Ø
376Ø RETURN
377Ø CLS:LOCATEØ,Ø:PRINT
     "ADVERTENCIA":LOCATEØ,
     1:PRINT"££££££££££"
378Ø LOCATEØ,12:PRINT"NO
     VALEN":LOCATE1Ø,
     1Ø:PRINT"NOMBRES
     PROPIOS":LOCATE1Ø,12:
     PRINT"PLURALES":
     LOCATE1Ø,14:
     PRINT"VERBOS" O
      TIEMPOS VERBALES"
379Ø LOCATEØ,22:PRINT
     "PULSA UNA TECLA..."
38ØØ A$=INKEY$:IFA$="
     "THEN 38ØØ
381Ø RETURN
382Ø N%=24Ø
383Ø SOUND7,&BØØ11111Ø
384Ø SOUND8,14
385Ø SOUNDØ,17Ø:
     SOUND1,Ø
386Ø FOR J=1 TO N%:NEXT
387Ø SOUNDØ,15Ø:
     SOUND1,Ø
388Ø FOR J=1 TO
     N%:NEXT
389Ø RETURN
```




GREEN BERET
THE GOONIES
SAMANTHA FOX
Strip Poker
PLUS
EXITOSERMA
Novedades
PARA TU MSX
NEMESIS
Knightmare
JAIL BREAK
NEMESIS
cartucho MSX - 5.200 ptas.
KNIGHTMARE
cartucho MSX - 5.200 ptas.
GOONIES
cartucho MSX - 5.200 ptas.
GREEN BERET
cartucho MSX - 5.200 ptas.
JAIL BREAK
cassette Amstrad - 2.200 ptas
disco Amstrad - 3.900 ptas
cassette Commodore - 2.200 ptas
cassette Spectrum - 2.000 ptas
cartucho MSX - 5.200 ptas
SAMANTHA FOX
cassette MSX - 1.995 ptas.
cassette Spectrum - 1.995 ptas.
cassette Amstrad - 1.995 ptas.
TAMBIEN DISPONIBLE EN KONAMI SHOP
LA UNICA TIENDA DEDICADA SOLO A VIDEOJUEGOS
C/. FRANCISCO NAVACERRADA, 19. MADRID
RECORTA Y ENVIA ESTE CUPON A SERMA: C/. CARDENAL BELLUGA, 21. 28028 MADRID - TLFS. 2562101-02
TITULO: SISTEMA:
NOMBRE Y APELLIDOS:
DIRECCION: COD. POSTAL:
POBLACION: PROVINCIA:
FORMA DE PAGO: POR TALON BANCARIO ☐ CONTRA REEMBOLSO ☐

¡PARTICIPA EN EL CONCURSO!

En INPUT estamos convencidos de que aún puedes hacer muchas más cosas con tu ordenador. Sin duda, muchos lectores estareis utilizando vuestro micro para funciones de lo más variadas, en unos casos; pintorescas, en otros; mientras que algunos listillos habrán podido utilizarlo para resolver tareas complejas. Es lógico, modificando programas y variando los periféricos nuestro ordenador puede prestar sus servicios en infinidad de facetas. INPUT quiere que esas aplicaciones y utilidades a las que has conseguido dedicar tu ordenador, sean conocidas por todos sus lectores y por eso ha organizado el «Concurso de Aplicaciones y Utilidades», en el que puede participar cualquiera de nuestros lectores.





BASES

UTILIDADES Y APLICACIONES: Si tu ordenador controla la calefacción de tu casa, gobierna un robot, dirige un pequeño negocio, organiza la maqueta de tu tren eléctrico, o cualquier cosa interesante u original; envíanos información gráfica y listados de tus programas, grabados en un cassette, diskette o microdrive.

Todo ello habrá de venir acompañado por un texto que aclare cuál es su objetivo, el modo de funcionamiento y una explicación del cometido que cumplen las distintas rutinas que lo componen. El texto se presentará en papel de tamaño folio y mecanografiado a dos espacios. No importa que la redacción no sea muy clara y cuidada; nuestro equipo de expertos se encargará de proporcionarle la forma más atractiva posible.

UN JURADO propio decidirá en cada momento qué colaboraciones reúnen los requisitos adecuados para su publicación, y evaluará la cuantía del premio en metálico al que se hagan acreedoras.

No olvideis indicar claramente para qué ordenador está preparado el material, así como vuestro nombre y dirección y, cuando sea posible, un teléfono de contacto. Entre todos los trabajos recibidos durante los próximos tres meses SORTEAREMOS:

- **Un premio de 50.000 ptas.**
- **Un premio de 25.000 ptas.**
- **Un premio de 10.000 ptas.**
  en material microinformático a elegir por los afortunados.

¡No os desanimeis!, por muy simples o complejas que puedan parecer vuestras ideas, todas están revisadas con el máximo interés.

INPUT MSX
Aribau, 185. Planta 1.ª
08021 BARCELONA

NOTA: INPUT no se responsabiliza de la devolución del material que no vaya acompañado por un sobre adecuado con el franqueo correspondiente.

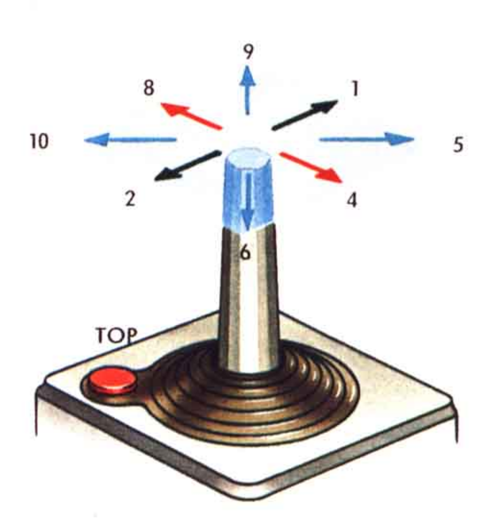
LA PROGRAMACION PARA JOYSTICKS

- COMO LEER LOS JOYSTICKS DESDE EL BASIC
- HAZ TUS JUEGOS MAS PROFESIONALES
- EL PULSADOR DE FUEGO

Los *joysticks* son la clave para tener unos juegos más profesionales. Pero el hacer que tus juegos resulten más divertidos no significa que tengas que aprender código máquina; puedes empezar con el BASIC.

Una diferencia evidente entre los juegos comerciales y los producidos en casa es con frecuencia la existencia de una opción para *joystick*. Sin embargo, no tienes que sumergirte en las profundidades intrincadas del código máquina para utilizar los *joysticks* en tus propios programas.

En esta sección del coleccionable, veremos la manera de utilizar los *joysticks* con programas escritos en BASIC, haciendo que tus programas de juegos resulten más profesionales y más divertidos de jugar.



En el próximo capítulo utilizaremos la rutina de *joystick* en un juego, por lo que no debes de olvidarte de almacenar el programa.

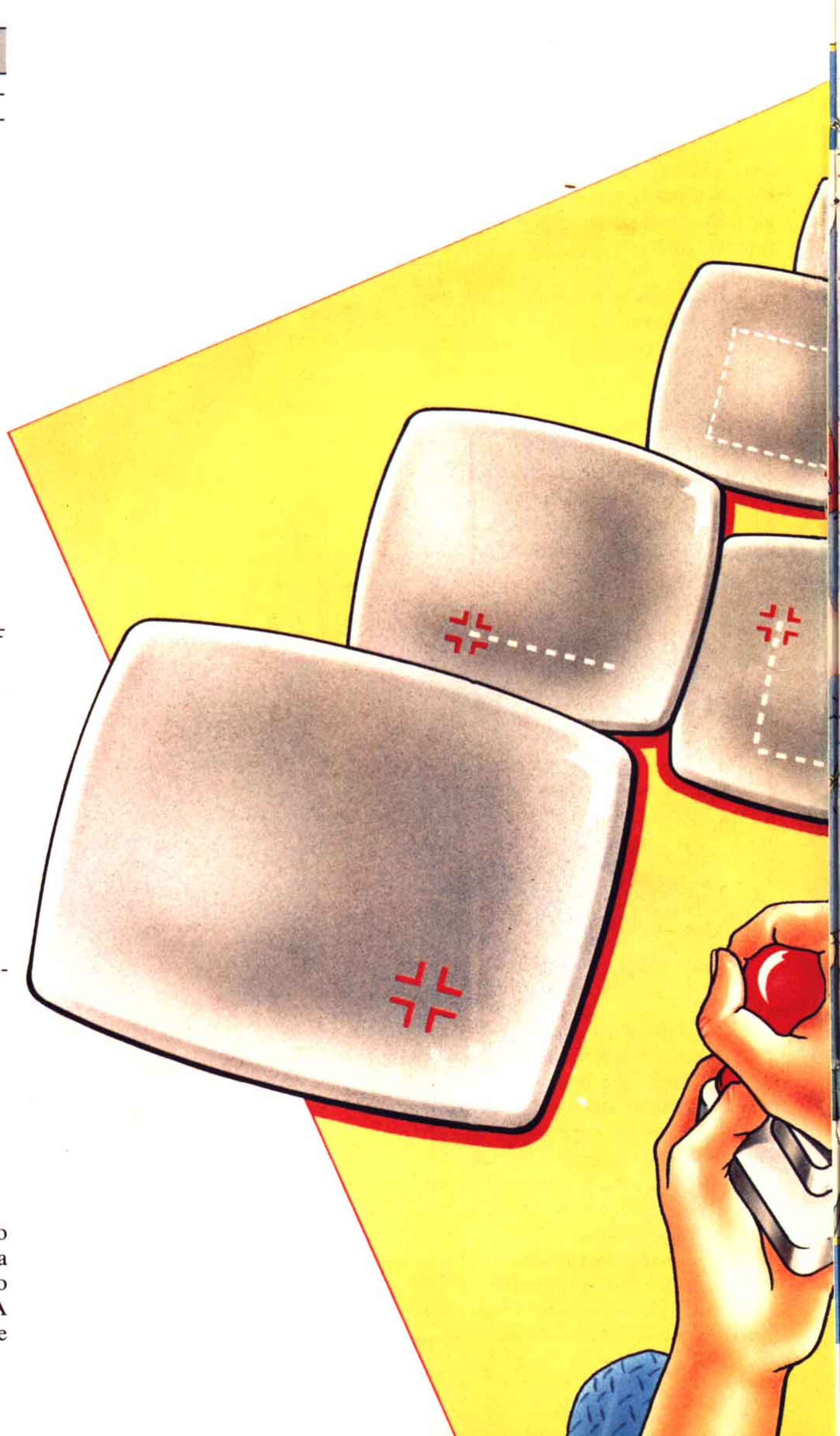
Lo primero que necesitas es un *joystick* adecuado. Todo programa está escrito para adaptarse a las características de los *joysticks* indicadas.

Existe una gran variedad de *joysticks* disponibles, pero todos están estandarizados para trabajar con vuestros ordenadores MSX.

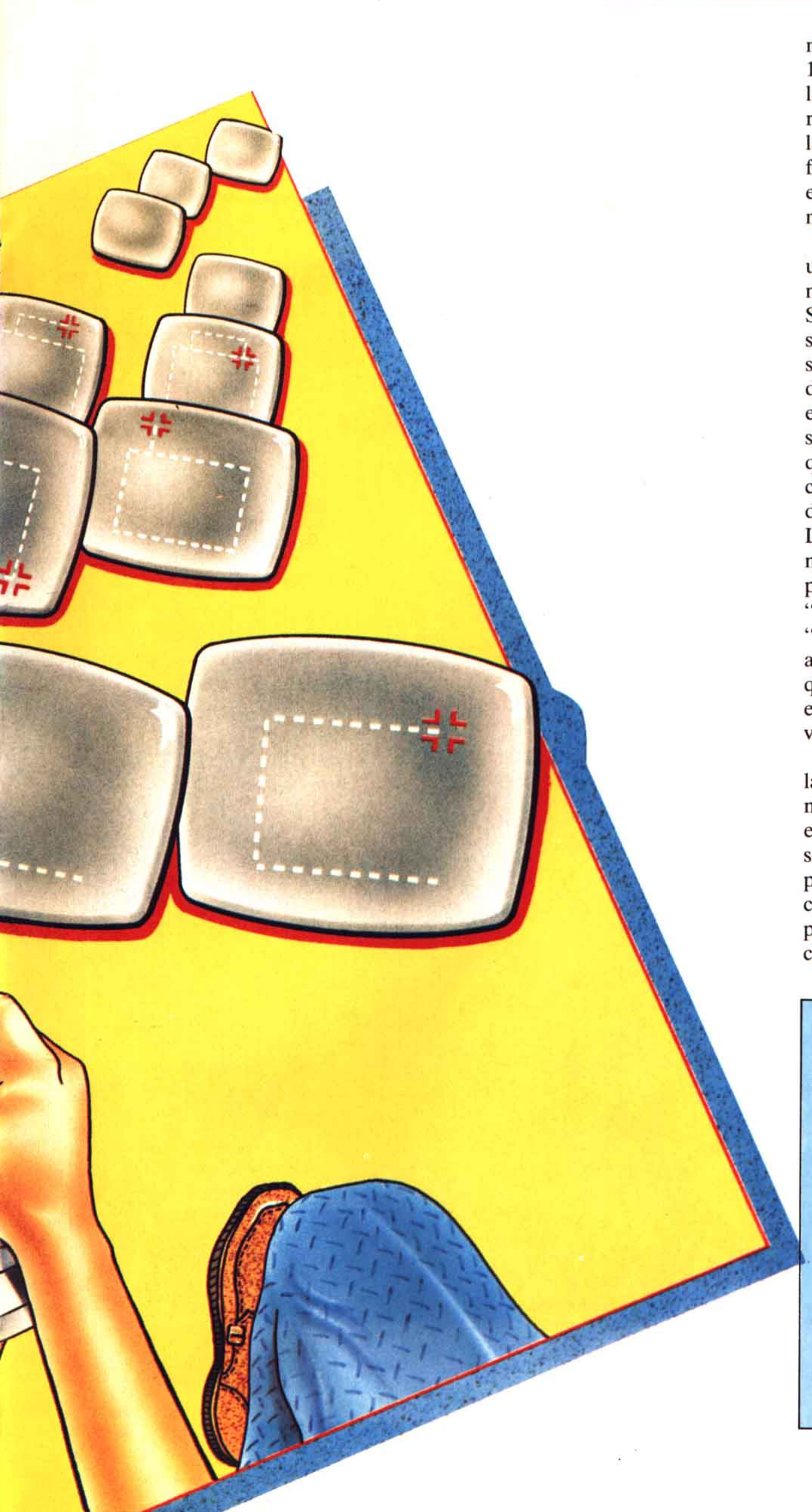
UN PUNTO DE MIRA

Teclea la siguiente sección de programa y tendrás un punto de mira controlable con tu *joystick*:

```
10 CLS:KEYOFF:COLOR5,1,
    1:SCREENØ,Ø,Ø
12Ø A=12Ø:B=1ØØ
13Ø A4$=CHR$(&HØ)
14Ø A5$=CHR$(&H1Ø)
15Ø A6$=CHR$(&H38)
16Ø A7$=CHR$(&H1Ø)
17Ø A8$=CHR$(&HØ)
25Ø A$=A4$+A5$+A6$+A7$
    +A8$
28Ø SCREEN 2,3,Ø
29Ø SPRITE$(Ø)=A$
47Ø LINE (Ø,Ø)-(256,45),7,BF
48Ø CIRCLE (15Ø,4Ø),2Ø,11,,,
    1.4
49Ø PAINT (15Ø,4Ø),11
5ØØ LINE (Ø,45)-(256,191),5,BF
51Ø S=STICK(Ø)
52Ø IF S=1 THEN B=B-1Ø
53Ø IF S=2 THEN
    A=A+15:B=B-10
54Ø IF S=3 THEN A=A+15
55Ø IF S=4 THEN
    A=A+15:B=B+1Ø
56Ø IF S=5 THEN B=B+1Ø
57Ø IF S=6 THEN A=A-
    15:B=B+1Ø
58Ø IF S=7 THEN A=A-15
59Ø IF S=8 THEN A=A-15:B=B-
    1Ø
6ØØ IF A=>25Ø THEN A=25Ø
61Ø IF A=<1Ø THEN A=1Ø
62Ø IF B=>181 THEN B=181
63Ø IF B=<1Ø THEN B=1Ø
64Ø PUT SPRITE Ø,(A,B),1,Ø
84Ø GOTO 51Ø
```



El programa empieza inicializando la pantalla en la línea 10, mientras la línea 120 define la posición del punto de mira que se empleará más tarde. A continuación se dibuja el punto de



mira creando un sprite entre las líneas 130 a 170; en la línea 280 preparamos la pantalla en modo de gráficos de alta resolución y de sprites ampliados; con las líneas 470 a 500 dibujamos un fondo que, aunque no es preciso para este programa, lo será cuando lo unamos con el de la próxima sección.

Una vez definido el sprite debemos utilizar un comando que controle nuestro *joystick*. Este comando es STICK. El número entre paréntesis selecciona el control desde el cursor; si es desde el primer *joystick* es 1; si es desde el segundo, es 2. De la línea 520 en adelante chequeamos todas las posiciones del *joystick*; cuando este chequeo detecta un cambio, el programa cambia los valores A o B encargados de almacenar la posición de la mira. Los valores que puede controlar el comando STICK y que dependen de la posición del *joystick* son: si detecta "1" arriba; "2" arriba a la derecha; "3" derecha; "4" derecha abajo; "5" abajo; "6" abajo a la izquierda; "7" izquierda; "8" izquierda arriba; cuando el *joystick* no esté en movimiento, el valor de STICK será "Ø".

Desde la línea 600 a 630 se controlan las posiciones de la mira para que no salga de la pantalla y aparezca por el lado opuesto. La línea 640 representa el punto de mira en pantalla, y, por fin, en la línea 840 devuelve el control del ordenador a la línea 510 para que chequee de nuevo las posiciones que puede adoptar el *joystick*.

P y R

¿Se le puede agregar una rutina de "joystick" a cualquiera de los juegos que aparecen en INPUT?

Sí, esto no encierra gran dificultad: simplemente hay que variar el núcleo del programa, el cual lee el teclado mediante la sentencia INKEY y sustituirlo por una rutina que lea el valor de STICK como la que hay en este programa y que está situada entre las líneas 52Ø y 54Ø.

EL JUEGO DE LA CAZA DE PATOS

Para los que no puedan esperar a que se levante la veda o no quieran tiritar de frío en el campo, aquí presentamos una rutina de disparo contra los patos, que podréis utilizar junto con la rutina de *joystick*.

Si has seguido atentamente el capítulo anterior, tendrás guardado en cinta un programa para mover por la pantalla el punto de mira. Pero aunque resulte satisfactorio disponer de un programa de este tipo escrito en BASIC, tal como está no vale para mucho.

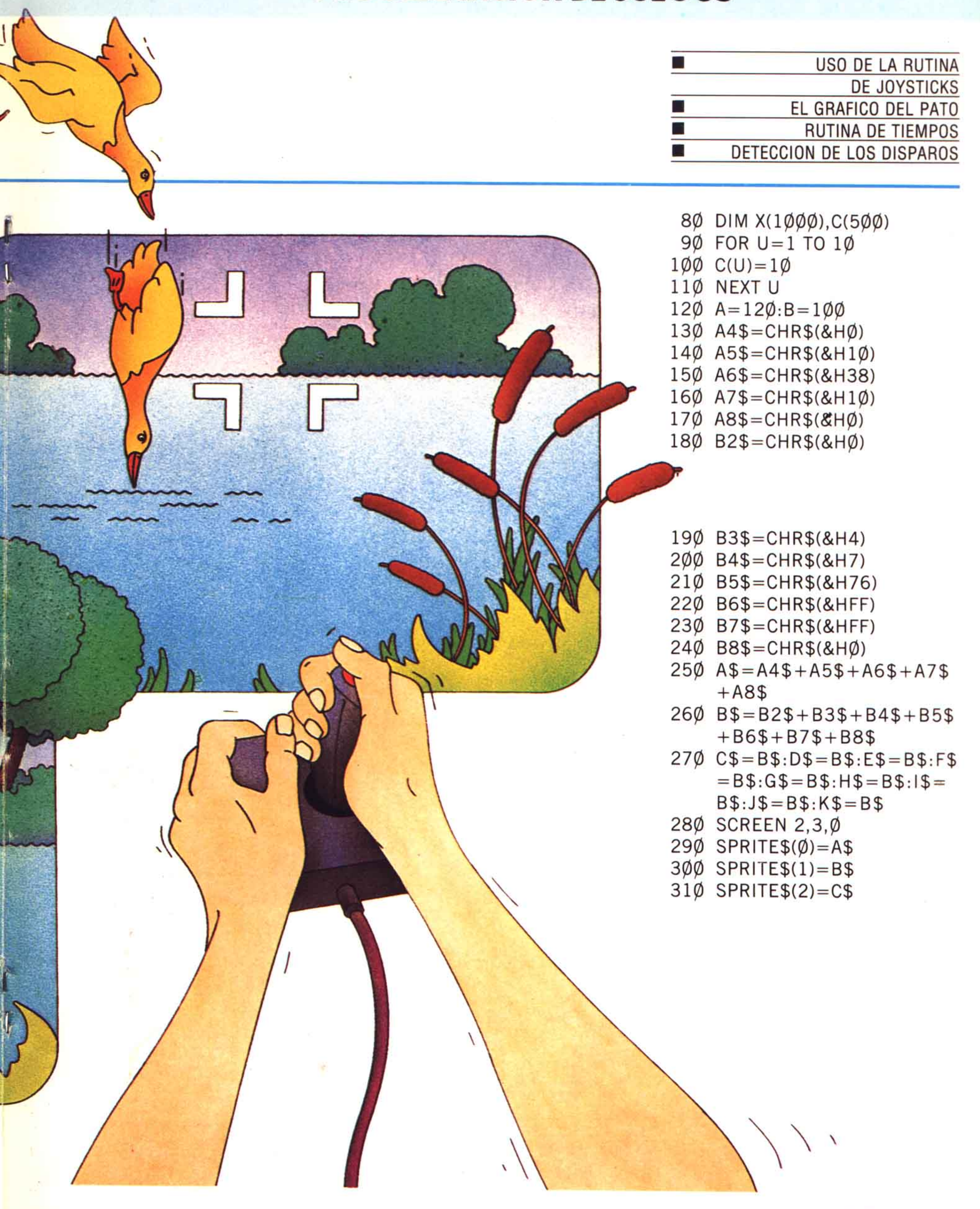
Por eso el siguiente paso es utilizar la nueva rutina dentro de un programa de juegos. Al añadir las siguientes líneas de programa especialmente escritas para tu máquina, tendrás un juego de caza de patos salvajes, aunque naturalmente puedes dibujar tus propios gráficos y disparar contra aviones, búfalos, dirigibles o cualquier otra cosa que te sugiera tu fantasía.

El objetivo del juego es abatir diez patos que aparecen en un corto intervalo de tiempo en posiciones aleatorias de la pantalla. El tanteo se basa en tu habilidad. Obtienes puntos por cada impacto, cuanto más rápido seas, más puntos obtendrás.

Además por cada tiro fallido se te quitarán puntos.

Carga en tu ordenador la rutina de *joystick* antes de teclear las nuevas líneas. A continuación añádele éstas y tendrás un programa de caza de patos con puntuación incorporada:

```
1Ø CLS:KEYOFF:COLOR5,1,
   1:SCREENØ,Ø,Ø
2Ø LOCATE8,8
3Ø PRINT "EL JUEGO DE LOS
   PATOS"
5Ø PRINT:PRINT:PRINT:PRINT:
   INPUT " ELIJA NIVEL DE
   JUEGO (1-8)";N
6Ø PLAY"V1ØL3ØDEGDEGCDF
   CDFDEGDEGCDFCDFFED"
7Ø CLS:LOCATE1Ø,13:PRINT"
   ESPERE UN MOMENTO"
```

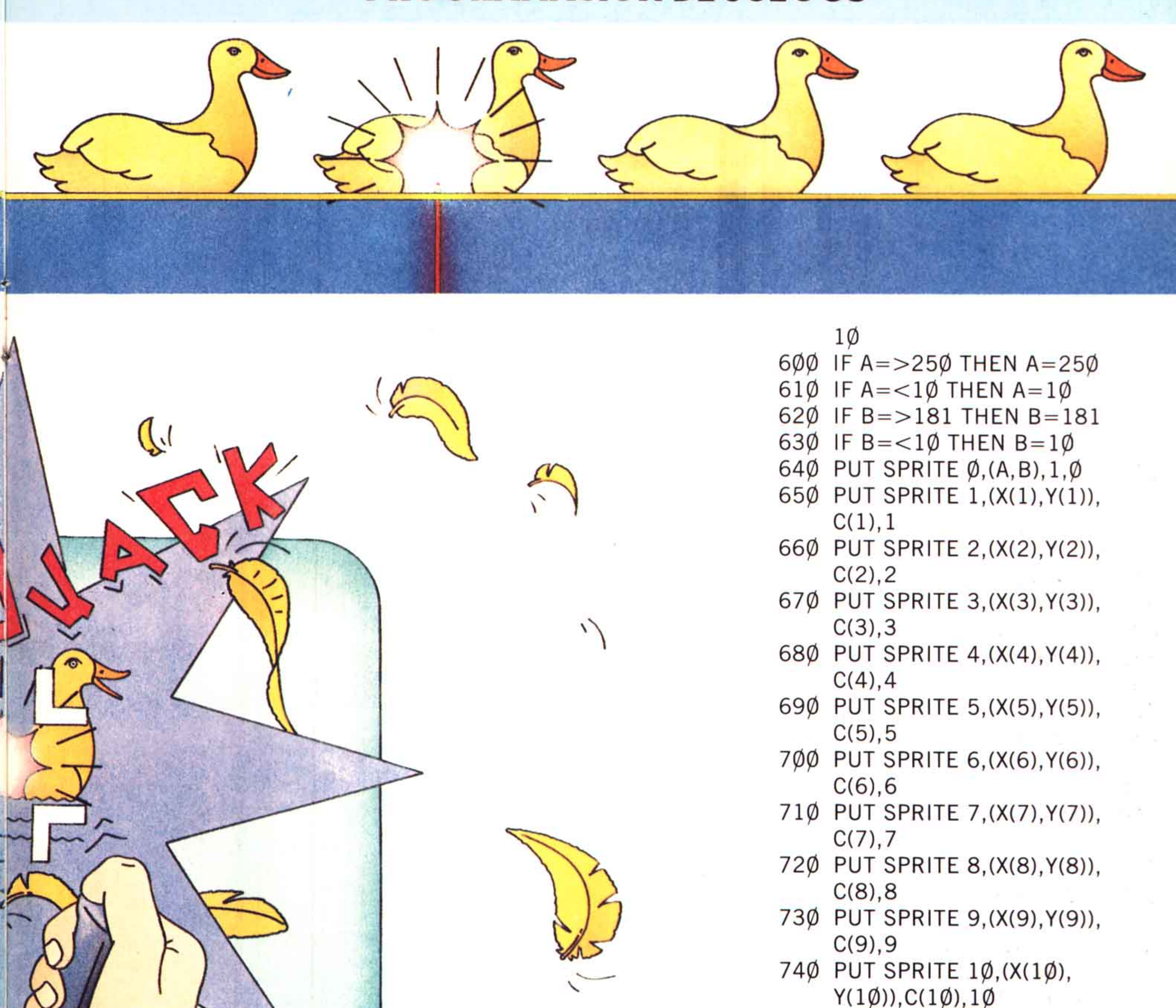


- USO DE LA RUTINA DE JOYSTICKS
- EL GRAFICO DEL PATO
- RUTINA DE TIEMPOS
- DETECCION DE LOS DISPAROS

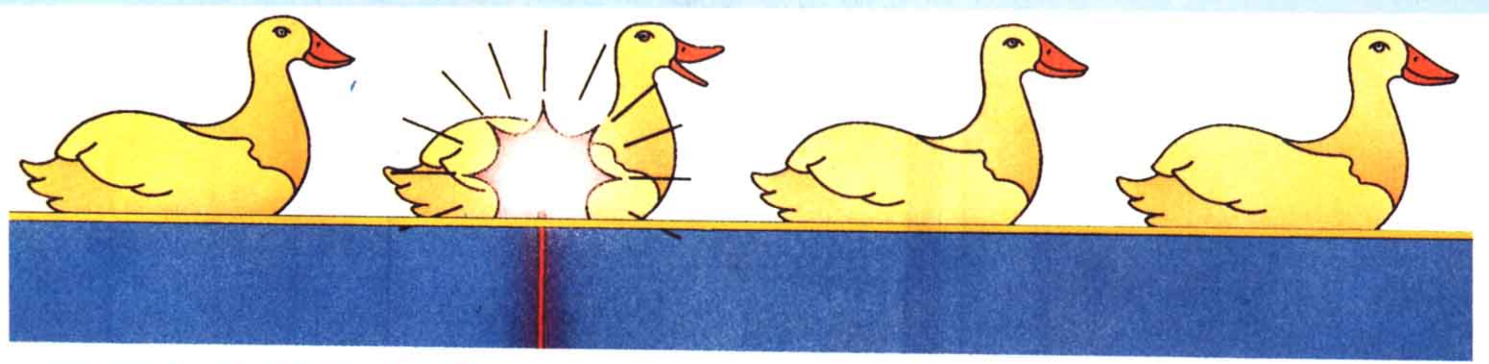
```
 8Ø DIM X(1ØØØ),C(5ØØ)
 9Ø FOR U=1 TO 1Ø
1ØØ C(U)=1Ø
11Ø NEXT U
12Ø A=12Ø:B=1ØØ
13Ø A4$=CHR$(&HØ)
14Ø A5$=CHR$(&H1Ø)
15Ø A6$=CHR$(&H38)
16Ø A7$=CHR$(&H1Ø)
17Ø A8$=CHR$(&HØ)
18Ø B2$=CHR$(&HØ)
19Ø B3$=CHR$(&H4)
2ØØ B4$=CHR$(&H7)
21Ø B5$=CHR$(&H76)
22Ø B6$=CHR$(&HFF)
23Ø B7$=CHR$(&HFF)
24Ø B8$=CHR$(&HØ)
25Ø A$=A4$+A5$+A6$+A7$
    +A8$
26Ø B$=B2$+B3$+B4$+B5$
    +B6$+B7$+B8$
27Ø C$=B$:D$=B$:E$=B$:F$
    =B$:G$=B$:H$=B$:I$=
    B$:J$=B$:K$=B$
28Ø SCREEN 2,3,Ø
29Ø SPRITE$(Ø)=A$
3ØØ SPRITE$(1)=B$
31Ø SPRITE$(2)=C$
```



```
320 SPRITE$(3)=D$
330 SPRITE$(4)=E$
340 SPRITE$(5)=F$
350 SPRITE$(6)=G$
360 SPRITE$(7)=H$
370 SPRITE$(8)=I$
380 SPRITE$(9)=J$
390 SPRITE$(10)=K$
400 VU=50
410 FORU=1TO10
420 X(U)=INT(RND(-TIME)*100)
430 Y(U)=VU:VU=VU+10
440 NEXT U
460 CLS
470 LINE (0,0)-(256,45),7,BF
480 CIRCLE (150,40),20,11,,,
    1.4
490 PAINT (150,40),11
500 LINE (0,45)-(256,191),5,BF
510 S=STICK(0)
520 IF S=1 THEN B=B+10
530 IF S=2 THEN
    A=A+15:B=B-10
540 IF S=3 THEN A=A+15
550 IF S=4 THEN
    A=A+15:B=B+10
560 IF S=5 THEN B=B+10
570 IF S=6 THEN A=A-
    15:B=B+10
580 IF S=7 THEN A=A-15
590 IF S=8 THEN A=A-15:B=B-
```



```
     10
600  IF A=>250 THEN A=250
610  IF A=<10 THEN A=10
620  IF B=>181 THEN B=181
630  IF B=<10 THEN B=10
640  PUT SPRITE 0,(A,B),1,0
650  PUT SPRITE 1,(X(1),Y(1)),
     C(1),1
660  PUT SPRITE 2,(X(2),Y(2)),
     C(2),2
670  PUT SPRITE 3,(X(3),Y(3)),
     C(3),3
680  PUT SPRITE 4,(X(4),Y(4)),
     C(4),4
690  PUT SPRITE 5,(X(5),Y(5)),
     C(5),5
700  PUT SPRITE 6,(X(6),Y(6)),
     C(6),6
710  PUT SPRITE 7,(X(7),Y(7)),
     C(7),7
720  PUT SPRITE 8,(X(8),Y(8)),
     C(8),8
730  PUT SPRITE 9,(X(9),Y(9)),
     C(9),9
740  PUT SPRITE 10,(X(10),
     Y(10)),C(10),10
750  FOR U=1 TO 10
760  X(U)=X(U)+N
770  IF X(U)=>256 THEN
     C(U)=0
780  NEXT U
```



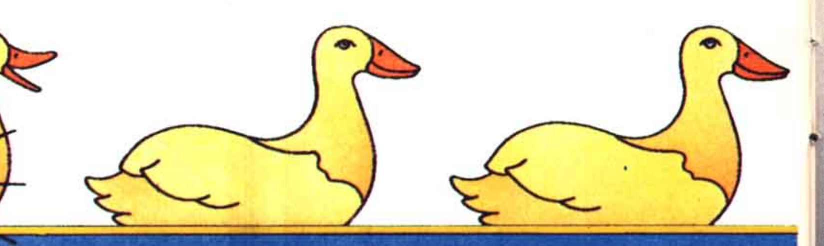
```
790 IFC(1)=0ANDC(2)=0ANDC
    (3)=0ANDC(4)=0ANDC(5)
    =0ANDC(6)=0ANDC(7)=0
    ANDC(8)=0ANDC(9)=0AN
    DC(10)=0THENGOTO990
800 SPRITE ON
810 ON SPRITE GOSUB 850
820 STRIG(0)ON
830 ON STRIG GOSUB 1080
840 GOTO 510
850 STRIG(0)ON
860 ON STRIG GOSUB 890
870 GOTO 840
890 SPRITE ON
900 ON SPRITE GOSUB 920
910 GOTO 800
920 UH=1:BEEP
930 FOR U=50 TO 140 STEP 10
940 IF B=U THEN C(UH)=0
950 UH=UH+1
960 NEXT U
970 PA=PA+1000*N
980 GOTO 800
990 SCREEN 0,0,0
1000 COLOR 8
1010 LOCATE6,10
1020 PRINT "PUNTUACION:";PA
1040 LOCATE 3,15:INPUT
     "OTRA VEZ?.. S/N";PR$
1050 IF PR$="S" OR PR$="s"
     THEN GOTO 90 ELSE
     1060
1060 CLS:KEYOFF:LOCATE 10,
     10:PRINT"ADIOS
     PARDILLOS"
1070 END
1080 BEEP
1090 RETURN
```

De la línea 20 a la 70 se crea una presentación en la que se nos preguntará el nivel de dificultad entre 1-8, con el que queremos jugar. La línea 80 dimensiona 2 variables que nos serán útiles más tarde. De la línea 90 a 110 se asigna el color a cada uno de los patos, mientras que desde la línea 180 a 240 se diseña un sprite con un sencillo dibujo de un pato. Desde la línea 300 a la 390 se asignan los números de sprites a cada uno de los patos. De la 400 a 450 se asignan las posiciones aleatorias de los patos en la pantalla. Las líneas comprendidas entre 650 y 740 se encargan de representar los diez patos en la pantalla. Desde la línea 750 a 790 se consigue que los patos avancen a una determinada velocidad y que vayan desapareciendo conforme llegan al final de la pantalla. Las líneas 800 y 810 son las encargadas de detectar si el sprite del punto de mira choca con el de algún pato; si esto ocurre el control pasa a la línea 850 y 860, donde se comprueba si se tiene pulsado el botón de disparo; esto se hace con el comando STRIG. El valor entre paréntesis selecciona el botón disparador; éste será la barra espaciadora cuando el valor sea 0; el botón disparador número 1 del *joystick* cuando el valor sea de 1 o 3, y el botón disparador número 2 del *joystick* cuando los valores sean 2 o 4. Si no se tiene pulsado el botón de disparo el control retorna al programa; si, por el contrario, lo tenemos pulsado, el control pasa a las líneas 890 y 900, donde se comprueba si el sprite del punto de mira aún está encima de alguno de los sprites de los patos. Si esto no es así el control retorna al programa; caso contrario se ponen en acción las líneas comprendidas entre 920 y 980, donde se produce el sonido BEEP del disparo, se localiza el pato que ha sido alcanzado y a éste se le atribuye el color 0 para que desaparezca. Después el control vuelve al programa. Es entonces cuando empieza una subrutina de fin de programa que se extiende desde la línea 990 hasta la línea 1070. Esta subrutina es llamada desde la línea 790.

P y R

¿Qué hay que hacer para cambiar el gráfico del pato por un blanco diferente?

Simplemente hay que alterar los valores de las variables de las líneas 180 a 240, y para variar el color basta alterar la línea 100.



MACROIMPRESION

DESCRIPCION DEL PROGRAMA

El programa presentado es un traductor de códigos para impresión de textos en macrotipos en la impresora, es decir, genera textos impresos de tamaño superior al normal en la impresora, pudiendo seleccionar la escala de ampliación.

El autor ha creado una aplicación que permite la escritura en la impresora tanto en sentido horizontal (modo normal), que puede ser empleado como encabezamiento de programas o trabajos listados por impresora, o bien, en sentido vertical, que será empleado para generar grandes textos propagandísticos o señalizaciones diversas.

El aumento se consigue leyendo en memoria los 8 bytes que componen cada carácter, haciendo uso para esto de la función *BASE(2)*, que nos da la dirección de comienzo de la tabla de caracteres multiplicada por 8 (longitud de cada carácter) y le sumamos el código *ASCII* del carácter en cuestión. Así calculamos la dirección inicial del carácter deseado.

IMPRESION VERTICAL

Localizado en las líneas 130 a 350, primeramente cambia el interlineado de la impresora a 1/8′ (24∅ LPRINT CHR$(27)"B") para unir los caracteres gráficos.



El bucle A (lin 250) se mueve a lo largo del texto a imprimir. El bucle B apunta al bit a comprobar, y el bucle C (lin 280) se mueve desde el último al primer byte de la letra para girarla 90°.

IMPRESION HORIZONTAL

Líneas 350 a 380. Se leen secuencialmente todos los bits de los ocho bytes del carácter, y en caso de encontrar algún «uno» se almacena en la cadena de trabajo C$. Una vez completado el carácter, se pasa la información a la cadena D$ para poder continuar con la siguiente letra. Una vez completada la cadena de entrada se imprime D$.

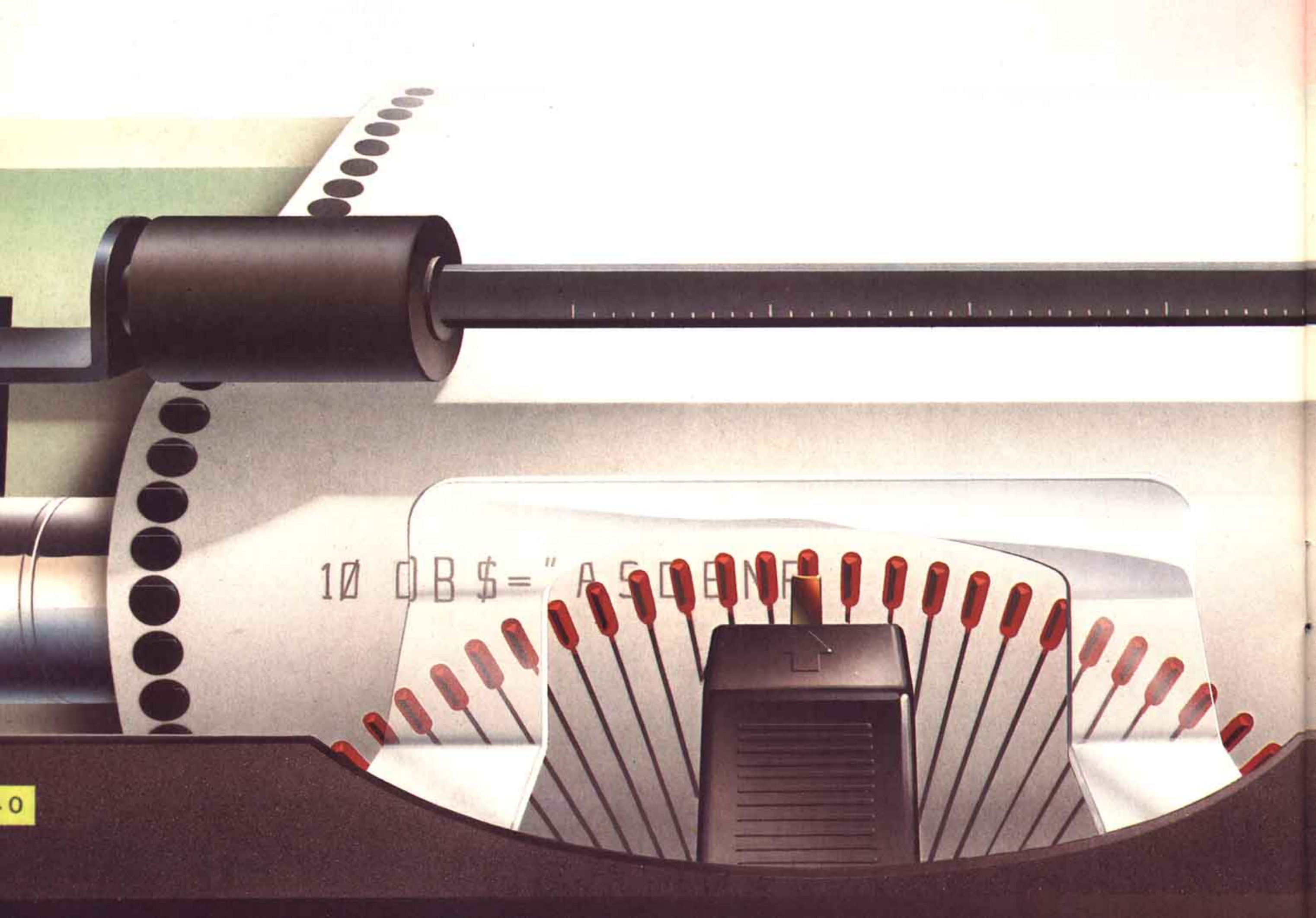
NOTAS DE INTERES

Se observará que para la construcción de la cadena B$ a partir de la lectura de memoria, se inicializa esta cadena con «ceros». Esto es debido a que la función BIN$ retorna un número binario, pero sólo da cifras a partir del primer «uno» de la izquierda (primer bit significativo), con lo que se hace necesario rellenar por la izquierda, y recuperar después los 8 dígitos del byte por la derecha.

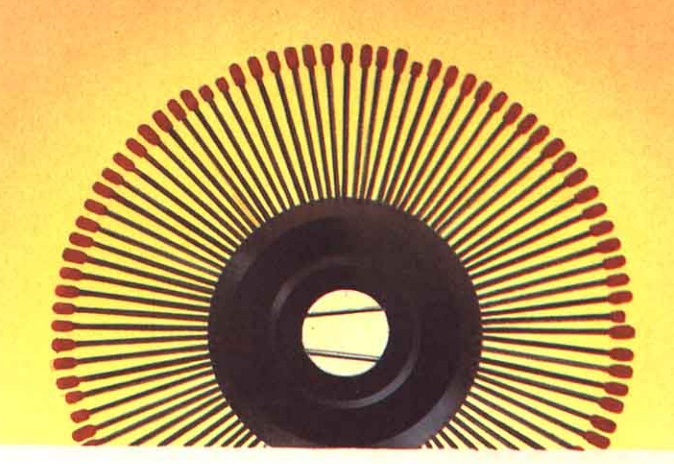
COMENTARIOS

El programa está bien resuelto, y la documentación aportada es bastante completa. Debemos recordar que se debe huir en lo posible del empleo de caracteres especiales dentro del listado del programa, que como es sabido crean problemas a la hora de sacar un listado por la impresora. Es aconsejable emplear caracteres estandard *ASCII*, o bien, incluir en el propio listado una definición clara del carácter especial empleado. Solamente así, otros lectores podrán entrar el programa en sus ordenadores con garantías de éxito en su funcionamiento, evitando las siempre tediosas puestas en marcha.

```
1 REM ******************
2 REM *                *
3 REM * PROGRAMA:
  MACRO-IMPRESION      *
4 REM *                *
5 REM * AUTOR: F.
  CARBALLO RUBIRA      *
6 REM *                *
7 REM *                *
8 REM *                *
9 REM ******************
10 CLS:CLEAR 400:DEFSTR I
20 LOCATE 8,
   1:PRINT"==========="
30 LOCATE 8,
   2:PRINT"=":LOCATE 26,
   2:PRINT"="
```



```
40 LOCATE 8,3:PRINT"=
   MACRO IMPRESION ="
50 LOCATE 8,
   4:PRINT"=":LOCATE 26,
   4:PRINT"="
60 LOCATE 8,
   5:PRINT"============
   70 LOCATE 2,
   8:PRINT"IMPRESION
   VERTICAL ...........1"
80 LOCATE 2,
   10:PRINT"IMPRESION
   NORMAL ...........2"
90 LOCATE 2,
   12:PRINT"AJUSTE
   IMPRESORA ...........3"
95 LOCATE 2,14:PRINT"FINAL
   DE PROCESO .......>ESC<"
100 I=INKEY$:IF I="" GOTO
   100 ELSE IF I="1" GOTO
   140 ELSE IF I="2" GOTO
   360
110 IF I=CHR$(27) THEN
   CLS:END
115 IF I="3" THEN GOSUB 700
120 GOTO 10
130 REM SUB.MACRO-
   IMP.VERTICAL
140 CLS
150 PRINT
160 INPUT "DAME TEXTO ";A$
170 PRINT
180 PRINT:INPUT "DAME
   CARACTER (ENTER=bla) ";
   Z$:PRINT:PRINT:IF Z$=""
   THEN Z$="é":REM Graf
   cuadr en blanco
190 INPUT "DAME ESCALA (1 a
   11)";N
200 IF N=0 THEN N=2
210 PRINT:PRINT
220 PRINT "QUIERES
   CENTRADO AUTOMATICO
   (S/N)"
230 I=INKEY$:IF I="" THEN
   230 ELSE IF I="S" OR
   I="s" THEN MAR=44-4*N
   ELSE PRINT: INPUT "DAME
```



```
     TABULADOR ";MAR
240  LPRINTCHR$(27)"B"
250  FOR A=1 TO LEN(A$)
260  B$=MID$(A$,A,1)
270  FOR B=1 TO 8
280  FOR C=7 TO 0 STEP-1
290  IF MID$(RIGHT$("000000
     00"+BIN$(VPEEK(BASE(2
     )+8*ASC(B$)+C)),8),B,1)
     ="1"THEN
     C$=C$+STRING$(N,Z$)
     ELSE C$=C$+SPACE$(N)
300  NEXT C
310  FOR U=1 TO
     NÑ2:LPRINTTAB(MAR)
     C$:NEXT U:C$=""
320  NEXT B
330  NEXT A
340  LPRINTCHR$(27)"A":GOSUB
     590:PRINT:GOTO 140
350  REM SUB.MACRO-
     IMP.HORIZONTAL
360  CLS:PRINT:INPUT"DAME
     TEXTO ";A$
370  PRINT
380  PRINT:INPUT"DAME
     CARACTER (ENTER=EL
     SUYO)";X$
390  PRINT:PRINT:INPUT"DAME
     ESCALA (ENTER=2) ";N
400  PRINT:PRINT
410  IF N=0 THEN N=2
420  'BUCLE QUE INVESTIGA EN
     LOS 8 BITS DE CADA LETRA
430  'IF LEN(A$)>8 GOTO 360
440  FOR A=0 TO 7
450  'BUCLE QUE SE MUEVE A
     LO LARGO DE LA CADENA
     PARA LEER LOS 8 BITS DE
     CADA UNO DE LOS 8 BYTES
     DE LA LETRA
460  FOR B=1 TO LEN(A$)
470  B$="00000000":E$=MID
     $(A$,B,1)
480  B$=B$+BIN$(VPEEK(BAS
     E(2)+8*ASC(E$)+A))
490  B$=RIGHT$(B$,8)
500  FOR S=1 TO 7
510  IF MID$(B$,S,1)="0" THEN
     C$=C$+SPACE$(N):GOTO
     530
```



```
520 IF X$="" THEN
    C$=C$+STRING$(N,E$)
    ELSE C$=C$+STRING$(N,
    X$)
530 NEXT S
540 D$=D$+C$:C$="":'A↓ADE
    CARACTERES A LA CADENA
    D$ Y LIMPIA LA CADENA DE
    TR ABAJO C$
550 NEXT B
560 FOR U=1 TO NÑ2:LPRINT
    D$:NEXT
    U:D$="":'IMPRIME Y
    LIMPIA D$
570 NEXT A
580 GOSUB 590:PRINT:GOTO
    360
590 LOCATE 8,20:PRINT"OTRA
    VEZ (S/N) "
600 I=INKEY$
610 IF I="S" OR I="s" THEN
    RETURN
620 IF I="N" OR I="n" OR
    I=CHR$(27) THEN RUN
630 GOTO 600
700 REM IMPRESORA
710 CLS:PRINTTAB(10)"
    IMPRESORA
720 PRINT:PRINT
730 PRINT"EXPANDIDA
    .........A"
740 PRINT"ELITE .........E"
750 PRINT"COMPRIMIDA
    .........Q"
760 PRINT"ITALICA .........I"
770 PRINT "SUBRAYADO
    .........X"
780 PRINT"RESETEO .........R"
790 PRINT:PRINT"SALIR....
    PULSAR ESC"
800 LOCATE 1,16:PRINT"ENTRE
    OPCION":LOCATE 15,16
810 I=INKEY$:IF I="" GOTO
    810
820 IF I="A" OR I="a" THEN
    LPRINTCHR$(14);
    :PRINT"EXPANDIDA "
830 IF I="E" OR I="e" THEN
    LPRINTCHR$(27)"E";
    :PRINT"ELITE "
835 IF I="I" OR I="i" THEN
    LPRINTCHR$(27)"4";
    :PRINT"ITALICA "
840 IF I="Q" OR I="q" THEN
    LPRINTCHR$(27)"Q";
    :PRINT"COMPRIMIDA"
850 IF I="X" OR I="x" THEN
    LPRINTCHR$(27)"X";
    :PRINT"SUBRAYADO "
860 IF I="R" OR I="r" THEN
    LPRINTCHR$(27)"c1";
    :LPRINTCHR$(7);
    :PRINT"RESETEO "
870 IF I=CHR$(27) THEN
    RETURN
880 GOTO 800
```

LOTO MSX

Muchos de los lectores de INPUT MSX habrán sentido en muchas ocasiones el deseo de «jugar» en algún juego de apuestas múltiples, como la quiniela futbolística o la quiniela hípica, y tal vez, desde hace poco más de un año, al «loto». Algunos, más interesados, tal vez empleen los denominados sistemas de reducción con la doble finalidad de maximizar las posibilidades reduciendo a su vez el gasto. Pues bien, el presente programa está hecho justamente para eso, para suprimir el tedioso trabajo de realizar la reducción de forma manual.

El programa que presentamos ahora solicita de entrada la cantidad de números sobre los cuales se desea jugar (línea 2Ø del programa; obvio es decir que este número jamás será igual o inferior a seis). En la siguiente línea nos solicita el nivel de reducción al cual deseamos jugar (5, 4 o 3).

Las líneas 50-100 son las encargadas de crear una matriz unidimensional en donde almacenamos los números concretos sobre los que deseamos jugar, y nos los solicita mediante un input inserto en un bucle.

Las líneas 120-190 y 350-400 son las encargadas de crear todas las combinaciones posibles de seis números con la cantidad de números que deseemos jugar. Esto da lugar a grandes cantidades de posibles apuestas que se ven reducidas por obra del procedimiento expresado en las líneas 200-310.

Una vez determinadas las combinaciones que nos resultan útiles, son almacenadas mediante las líneas 330-340 en la matriz que previamente hemos dimensionado en la línea 130 y que tú puedes alterar en función de la disponibilidad de memoria que poseas y del número de combinaciones que preveas que puedan resultar.

La subrutina llamada desde la línea 410 se encarga, una vez realizado todo el procedimiento, de ajustar la matriz de combinaciones resultante a una matriz de expresión directa de las combinaciones que has de jugar en función de los números que tú indicaste en las líneas 50-100.

```
10 CLS:KEYOFF:WIDTH 4Ø
2Ø INPUT "Cuántos números";N
3Ø INPUT "Reducción al ";R
4Ø CLS
5Ø LOCATE 4,1Ø:PRINT "INTRODUCIR LOS NUMEROS POR ORDEN"
6Ø DIM H(N-1)
7Ø FOR G=Ø TO N-1
8Ø INPUT "Numero a jugar: ";N1
9Ø H(G)=N1
1ØØ NEXT G
11Ø CLS
12Ø LOCATE 1Ø,12:PRINT "ORDENADOR PROCESANDO"
13Ø DIM M(1ØØ,5):CO=Ø:CR=Ø
14Ø FOR A=1 TO N-5
15Ø FOR B=A+1 TO N-4
16Ø FOR C=B+1 TO N-3
17Ø FOR D=C+1 TO N-2
18Ø FOR E=D+1 TO N-1
19Ø FOR F=E+1 TO N
2ØØ CR=Ø
21Ø FOR K=CO TO Ø STEP -1
22Ø FOR T=Ø TO 5
```

```
23Ø IF A=M(K;T=+) THEN GOTO 44Ø
24Ø IF B=M(K,T) THEN GOTO 44Ø
25Ø IF C=M(K,T) THEN GOTO 44Ø
26Ø IF D=M(K,T) THEN GOTO 44Ø
27Ø IF E=M(K,T) THEN GOTO 44Ø
28Ø IF F=M(K,T) THEN GOTO 44Ø
29Ø NEXT T
3ØØ CR=Ø
31Ø NEXT K
32Ø REM Combinación válida
33Ø M(CO,Ø)=A:M(CO,1)=B:M(CO,2)=C:M(CO,3)=D:M(CO;4)=E:M(CO,5)=F
34Ø CO=CO+1
35Ø NEXT F
36Ø NEXT E
37Ø NEXT D
38Ø NEXT C
39Ø NEXT B
4ØØ NEXT A
41Ø GOSUB 46Ø
42Ø PRINT "MUCHA SUERTE"
43Ø END
44Ø CR=CR+1:IF CR=>(6-R) THEN GOTO 35Ø
45Ø GOTO 29Ø
46Ø DIM MF(CO-1,5)
47Ø FOR G=Ø TO N-1
48Ø FOR P=Ø TO CO
49Ø FOR Q=Ø TO 5
5ØØ IF M(P,Q)=G+1 THEN MF(P,Q)=H(G) ELSE GOTO 52Ø
51Ø M(P,Q)=Ø
52Ø NEXT Q
53Ø NEXT P
54Ø NEXT G
55Ø CLS
56Ø LOCATE 1Ø,2:PRINT " COMBINACIONES A JUGAR"
57Ø PRINT:PRINT:PRINT
58Ø FOR L=Ø TO CO-1
59Ø PRINT MF(L,Ø);TAB(5);MF(L,1);TAB(5),MF(L;2);TAB(5);MF(L,3);TAB(5);MF(1,4);TAB(5);
    MF(L,5),
6ØØ NEXT L
61Ø RETURN
```

Como podrán apreciar los lectores, el programa tiene como principal característica su gran simplicidad, pero por esta misma razón resulta excesivamente lento de procesar; por ello mismo el artículo queda abierto a la colaboración de los lectores que quieran enviar sus ideas a la redacción de INPUT aportando las posibles mejoras que se les ocurran.

Para dar una posible idea de las dificultades con que los usuarios pueden encontrarse os daremos unos pocos ejemplos.

El ciclo expuesto en las líneas 140-400 tiene la necesidad de examinar si deseáis combinar 12 números un total de 924 apuestas diferentes. Este número se eleva a 5005 si los números a combinar son 15, y llega hasta la cifra de 38760 si deseáis combinar 20 cifras. Todo ello os dará

una idea de las dificultades en lo relativo a tiempo de procesamiento que encontraréis si no os es posible encontrar unas rutinas que aceleren la ejecución.

Por otro lado, y para indicaros alguna ventaja, os puedo decir que una apuesta de diez números que en directo tiene un precio de 10.500 pesetas queda reducida a sólo dos apuestas con un coste de 100 pesetas; las 924 apuestas de 12 números se ven reducidas a sólo 4 apuestas y las 5005 de 15 números se quedan en únicamente 7 apuestas a realizar.



Puedes ganar 25.000 pts. o tu juego preferido

Encuesta para conocer tu perfil y preferencias

Hace ahora un año, la redacción de INPUT se inundó de encuestas enviadas por vosotros, los lectores. Gracias a esa entusiasta acogida fue posible conocer a fondo vuestras inquietudes y ajustar a ellas los contenidos de la revista.
Ahora, esta nueva encuesta pretende saber hasta qué punto hemos conseguido ese objetivo, y enterarnos de por dónde van ahora vuestras exigencias, siempre pensando en que INPUT siga siendo una herramienta indispensable para vuestros primeros escarceos por el mundo de la informática.

1. ¿Tienes ordenador? Sí ☐ No ☐
2. En caso afirmativo. ¿Cuál? ....
   Marca .... Modelo ....
3. ¿Cuánto tiempo hace que lo posees? ....
4. ¿Qué periféricos y accesorios tienes? ....
5. ¿Desde qué número lees INPUT? ....
6. ¿Con qué frecuencia la adquieres?
   TODAS ☐ DE VEZ EN CUANDO ☐ OTROS ☐
7. ¿Cuántas personas leen tu ejemplar de INPUT?
   Sólo yo ☐ Yo y mis amigos ☐
   Mi hermano, mi padre y yo ☐ Mi hermano y yo ☐
8. ¿Lees toda la revista? Sí ☐ No ☐
9. ¿Tecleas los programas que publicamos?
   Todo ☐
   Casi todo ☐
   Alguna cosa ☐
   Nada ☐
10. Califica qué secciones te gustan más con mayor puntuación (10 a 0)
    Programación ....
    Código Máquina ....
    Robótica ....
    Coleccionable ....
    Revista Software ....
    Aplicaciones ....
    Actualidad ....
    Libros ....
    Buzón ....
    Educación ....
    Inteligencia Artificial ....
    El micro Z-80 ....
11. ¿Qué secciones o temas echas de menos en INPUT? ....
12. ¿Crees que alguna sección debe ampliarse? ¿Cuál?
    ....
13. ¿Cuál debería reducir su extensión? ....
14. ¿Tienes en cuenta los comentarios sobre juegos (Revta. de Software) a la hora de adquirirlas en el mercado?
    Al 100 % ☐ En gran medida ☐ Poco ☐ Nada ☐
15. ¿De todos los artículos que has leído en INPUT, nos podrías destacar algunos?
    N.º 1 ....
    N.º 2 ....
    N.º 3 ....
    N.º 4 ....
    N.º 5 ....
    N.º 6 ....

SOBRE TI

16. Sexo:
    M ☐ F ☐
    Edad:
17. Hasta 12 años ☐ De 21 a 25 años ☐
    De 12 a 14 años ☐ De 26 a 35 años ☐
    De 15 a 17 años ☐ De 36 a 45 años ☐
    De 18 a 20 años ☐ Más de 45 años ☐
18. ¿Cuál es tu situación?
    Estudio enseñanza básica ☐
    Estudio enseñanza media ☐
    Estudio enseñanza Universitaria ☐
    Estoy parado ☐
    Trabajo (en este último caso responde a la siguiente pregunta) ☐
19. ¿Cuál es tu profesión? ....
20. ¿Con qué objetivo adquiriste tu ordenador?
    Jugar ☐ Educativo ☐ Música ☐
    Utilidades ☐ Profesionales ☐ Dibujo ☐
21. ¿Cuál es tu nivel de conocimientos de informática?
    0 ☐ Bastante ☐ Aficionado con exp. ☐
    Profesional ☐
22. Aparte de la informática, ¿qué otras aficiones tienes?
    Deporte ☐ Cine ☐ Música ☐ ☐
    Vídeo ☐ Pintura ☐ Teatro ☐ ☐
    Motociclismo ☐ Fotografía ☐ ☐ ☐
23. ¿Estás satisfecho con tu ordenador? ....
24. En caso de que te tuvieses que comprar otro, ¿por cuál te decidirías? ....
25. ¿Lees otras revistas de informática? Sí ☐ No ☐
26. ¿Cuáles? ....

Entre los que contestéis a esta encuesta, se sorteará un premio en metálico de 25.000 pts. y diez programas a elegir por los agraciados. Así, que escribe tus datos para participar. La fecha límite es el próximo 5 de marzo. No es preciso que contestes las preguntas para las que no tengas respuesta. Tampoco es necesario que las respuestas sean favorables a nuestra labor para que puedas participar en el sorteo. Critícanos si honestamente crees que debes hacerlo. Envíanos este cuestionario o su fotocopia a: ENCUESTA INPUT MSX, Aribau, 185, 1.º - 08021 Barcelona

Nombre .... Apellidos ....
Edad .... Dirección .... Dto. Postal ....
Teléfono de contacto .... Gracias y suerte con el premio.

NUMEROS BAJO CERO

- LOS NUMEROS NEGATIVOS SE HACEN NECESARIOS
- PROGRAMA DE CONVERSION PARA NUMEROS NEGATIVOS
- LA CONVENCION DEL SIGNO

La comprensión de los números binarios y hexadecimales no suele ser difícil, pero la expresión de valores negativos con estos sistemas de numeración puede presentar problemas.

En la programación de algunos juegos puede ser necesario utilizar valores negativos, por ejemplo, para desplazarse sobre la pantalla en determinadas direcciones. Las direcciones van codificadas en grupos de ocho bits, ya que esa es la única manera en que el ordenador puede almacenar los datos en su memoria. Pero surge un problema: como hemos visto, un byte puede representar cualquier número desde 0 hasta 255, es decir, desde 0000 0000 hasta 1111 1111. Pero con esto se agotan las posibilidades de un número binario de ocho bits, que ya no tiene sitio para un signo menos o más, ni forma alguna de representarlo.

En la aritmética ordinaria, al restar 1 de 0 se obtiene el valor −1; aquí tienes ese mismo cálculo realizado con números binarios de ocho bits:

```
0000 0000

       - 1
---------
1111 1111
```

Puedes probar a tomar un 1 prestado de la columna situada más a la izquierda, pero cuando el número binario está limitado a ocho bits no hay manera de hacer nada. Análogamente, al restar otro 1 (en la aritmética tradicional se obtendría como resultado un −2) tenemos 1111 1110, que en binario corresponde al número decimal 254, ya que 1111 1111 es 255.

Esto no son curiosidades típicas de los números binarios: imagínate por ejemplo lo que ocurriría en decimal si sólo hubiese tres sitios o columnas en las que poner los números. Mira lo que sucedería si se intentase sumar 100 a 999 con estas limitaciones:

```
    100

  + 999
-------
 (1)099
```

El uno que figura en la columna de las unidades de mil, está escrito entre paréntesis; en un sistema aritmético con sólo tres columnas no hay espacio para las que «te llevas». En un sistema de sólo tres lugares, el resultado de esta suma es pues 99, o sea, exactamente el resultado de 100 menos 1.

De la misma forma, sumar 998 con 100 en un sistema de tres lugares equivale a restar 2, mientras que restar 998 de 100 equivale a sumarle 2.

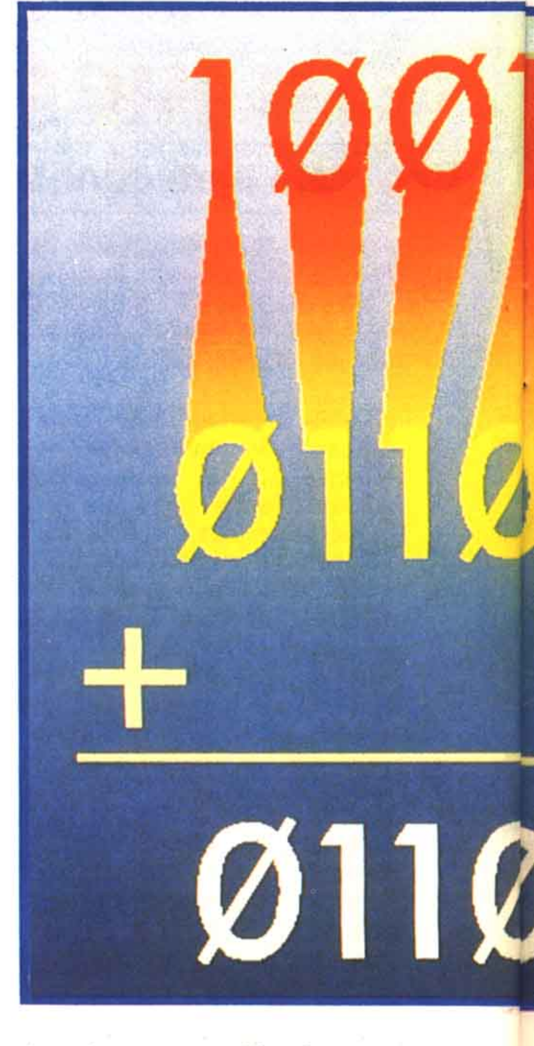
En conclusión, una misma sucesión de números de un byte puede representar un número positivo o negativo, sin tener en cuenta la posible confusión o dificultad en el cálculo.

¿Qué hacer entonces? En la mayoría de las aplicaciones de los ordenadores domésticos, este problema no suele plantearse: tanto las direcciones de memoria como los códigos de operación, ambos expresados en binario, pueden considerarse siempre como positivos. Las únicas ocasiones en que se encuentran números negativos son en los datos o en los saltos del código máquina, equivalentes a los GOTOs del BASIC.

CAMBIANDO LOS BITS

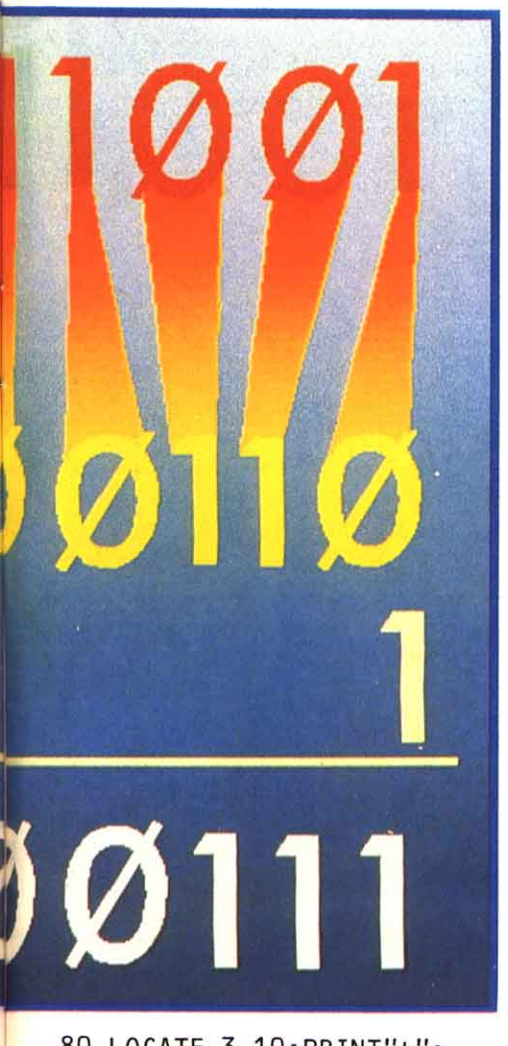
Para pasar de un valor en binario positivo a su correspondiente negativo, hay que aplicar el llamado procedimiento del complemento a 2; no tiene una base teórica de fácil comprensión, pero sin embargo funciona bien.

Para pasar desde un número en binario a su correspondiente valor negativo, hay que cambiar los ceros por unos y viceversa, sumando al final 1 al resultado.



El siguiente programa realiza precisamente esta función. Observa que cuando el número en binario está limitado a ocho cifras, su equivalente en hexadecimal consta exactamente de dos cifras: esto quiere decir que también el equivalente hexadecimal del complemento a 2 es el correspondiente valor negativo.

```
10 CLS:LOCATE 3,0:PRINT
   "NUMEROS EN COMPLEMENTO A
   2"
20 LOCATE 1,2:PRINT"DEC";
   TAB(14);"BIN";TAB(28);
   "HEX"
50 LOCATE 6,7:PRINT"........
   .........."
70 LOCATE 8,8:PRINT
   "COMPLEMENTO A 1"
```



```
80 LOCATE 3,10:PRINT"+":
   LOCATE 30,10:PRINT"+"
90 LOCATE 8,15:PRINT
   "COMPLEMENTO A 2"
92 LOCATE 0,18:  PRINT"------
   --------------------------
   ----"
94 LOCATE 8,19:PRINT
   "0 0 0 0 0 0 0 0"
95 C=0
99 REM*** IMPRESION VALORES
100 DD=-C
110 LOCATE 0,4:PRINT USING
    "####";C
115 LOCATE 0,17:PRINT USING
    "####";DD
120 Z=C:GOSUB 310:LOCATE
    29,4:PRINT A$
130 Z=DD:GOSUB 310:LOCATE
    29,17:PRINT A$
140 Z=C:GOSUB 320:F=4:GOSUB
    330
150 Z=DD:GOSUB 320:F=17:
    GOSUB 330
160 Z=NOT(C):GOSUB 320:F=10:
    GOSUB 330
230 B=STICK(0):B$=INKEY$:
    IF B=0 AND B$=""
    THEN 230
240 IF B=1 THEN C=C-1:
    IF C=-129 THEN C=127
    :BEEP
250 IF B=5 THEN C=C+1:
    IF C=128 THEN C=-128
    :BEEP
260 IF B$=" " THEN LOCATE
    0,20:INPUT C:LOCATE 0,20:
    PRINT"                 "
265 IF B$<>" " AND B<>1 AND
    B<>5 THEN GOTO 230
270 GOTO 100
300 REM*** SUBRUTINAS
310 A$=STRING$(2-LEN(RIGHT$
    (HEX$(Z),2)),"0"):A$=A$+
    RIGHT$(HEX$(Z),2):
    RETURN
320 A$=STRING$(8-LEN(RIGHT$
    (BIN$(Z),8)),"0"):A$=A$+
    RIGHT$(BIN$(Z),8):
    RETURN
330 FOR A=1 TO 8
340 LOCATE 6+2*A,F:PRINT
    MID$(A$,A,1)
350 NEXT A:RETURN
```

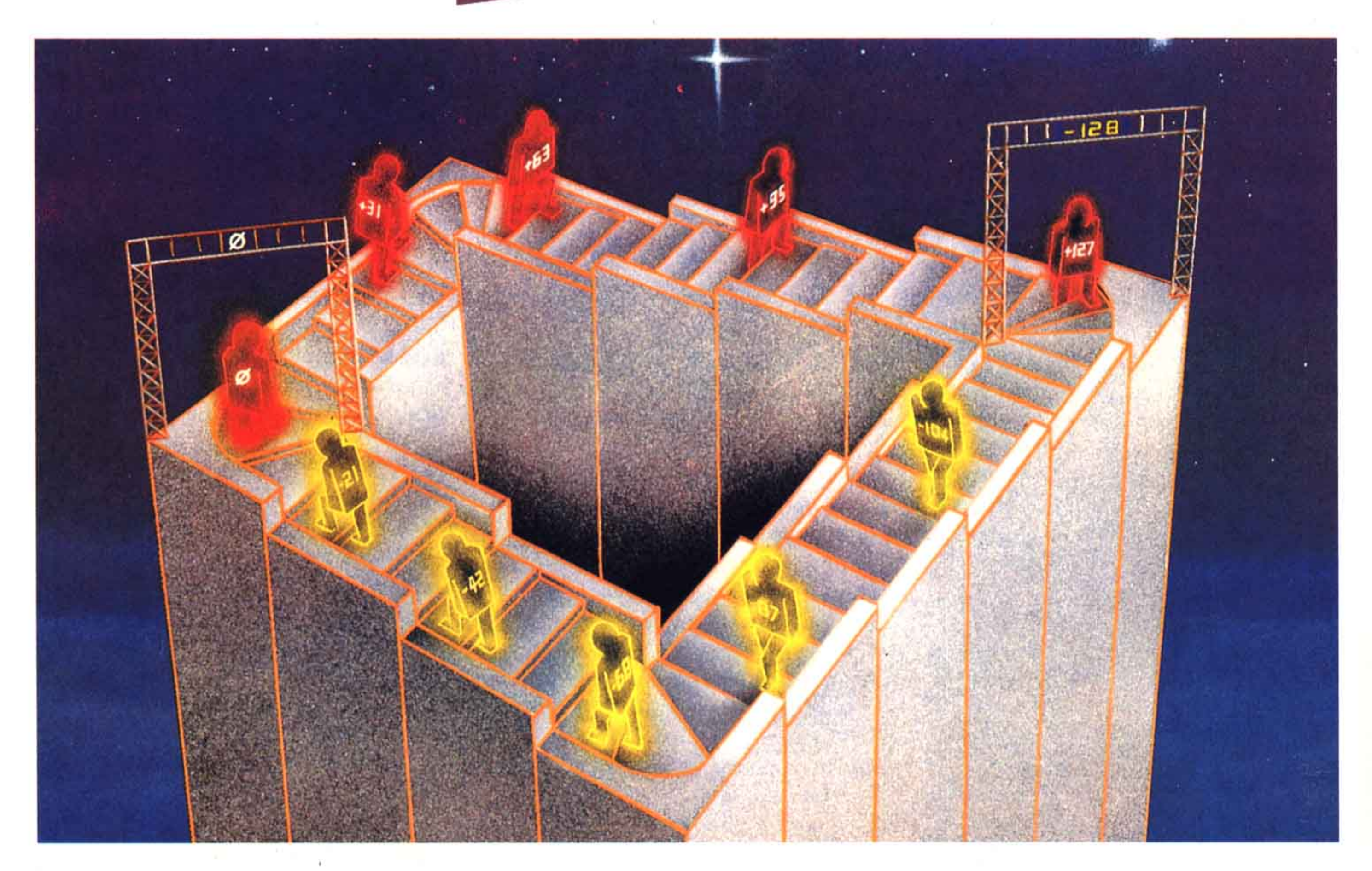
El programa consta de tres partes bien diferenciadas. La primera, desde la línea 10 hasta la 95, se ocupa de presentar en pantalla los encabezamientos de las distintas columnas de números. Cuando lo ejecutes verás en la línea superior la representación decimal, binaria y hexadecimal de un número. Hacia la mitad de la pantalla aparece la representación binaria del complemento a 1 del citado número. Este complemento a 1 se obtiene sin más que cambiar los ceros por unos, y viceversa, en el número binario original. También podemos considerar que el complemento a 1 de un valor binario A cualquiera, no es más que el resultado de NOT(A). Por último, en la línea inferior de la pantalla se representa el complemento a 2 del número, en decimal, en binario y en hexadecimal. El complemento a 2 se obtiene sin más que sumar 1 al valor del complemento a 1.

La segunda parte del programa, entre las líneas 99 y 160, se ocupa de llevar a cabo los cálculos necesarios y de presentar los valores actualizados en la pantalla. El cálculo de los valores del complemento a 1 y del complemento a 2 se lleva a cabo utilizando al-

GANADORES DE LOS MEJORES DE INPUT MSX

En el sorteo correspondiente al número 14 entre quienes escribisteis mandando vuestros votos a LOS MEJORES DE INPUT han resultado ganadores:

NOMBRE	LOCALIDAD	JUEGO ELEGIDO
Marino López García	Oviedo	Le Mans II
David Muñoz Paez	Barcelona	Green Beret
Óscar Larrea García	Almería	Hyper Rally
Jordi Cadellans Bausili	Barcelona	Green Beret
Prudencio Fuente Reizábal	Casetas (Zaragoza)	Flight Deck
Carlos Murillo Blasco	S. Boi de Llobregat (Barna)	Goonies
Javier Galán de Arribas	Barcelona	Nemesis
Pablo Zúmel Vaquero	Burgos	International Karate
Julián Santos Alcón	Cáceres	Green Beret
Miguel Ferreiro Lorenzo	Vigo (Pontevedra)	Jack de Nipper



gunas de las funciones del BASIC **MSX** como por ejemplo el operador lógico NOT y las funciones de conversión de base HEX$ y BIN$. En esta parte del programa se hacen llamadas a las subrutinas de las líneas 300 a 350, que se ocupan del manejo de las cadenas de caracteres con los valores binarios y hexadecimales. Es necesario trocear y manipular estas cadenas mediante las funciones LEFT$, RIGHT$ y MID$, para obtener una correcta presentación en pantalla.

La última parte del programa, comprendida entre las líneas 230 y 270, se ocupa de la entrada de datos desde el teclado. Cuando ejecutes el programa, verás aparecer en pantalla la representación en complemento a 1 y complemento a 2 del número cero. Para avanzar en el sentido de los números positivos no tienes más que pulsar la tecla de CURSOR ABAJO. Por el contrario, para los números negativos tienes que pulsar la tecla de CURSOR ARRIBA. Pulsando estas teclas podrás aumentar o disminuir los números de uno en uno. Si quieres ir más deprisa sólo tienes que pulsar la barra de espacios y a continuación teclear el número cuyos complementos deseas conocer.

LA CONVENCION DEL SIGNO

Como ya hemos dicho, para muchos fines un número en binario o en hexadecimal corresponde perfectamente a un número positivo o negativo. Sin embargo hay ocasiones en que se quiere conocer si un número es positivo o negativo.

Los saltos realizados en los programas en lenguaje máquina requieren que se especifique el número de bytes que abarcan, en valores positivos si el salto es hacia adelante y negativos si el salto es hacia atrás. El ordenador examina el primer bit del número en binario y determina por sí solo si se trata de un número positivo o negativo. Si el primer bit es un 1, el ordenador considera el número como negativo, mientras que si es 0 lo toma como positivo.

Este procedimiento se ajusta a una determinada convención de signos. Esto significa que el ordenador, en vez de manejar números binarios que varían entre 0 y 255, considera que su margen de variación va desde −128 a +127. En el programa de conversión en complemento a 2, habrás oído la señal de «bip» al llegar a 128. Esto se debe a que 128, que en binario es 1000 0000 y en hexadecimal es 80, corresponde a un valor negativo. (Haz la prueba: −1000 0000 + 1000 0000 = (1)0000 0000, que es 0 en binario de ocho bits; 80 + 80 = (1)00, que es 0 en hexadecimal de dos cifras; 128 − 128 = 0 en decimal.)

¿Qué es lo que sucede entonces? Dado que 1000 0000 tiene un 1 en su primer bit, el ordenador lo considera como un número negativo, el −128. Por otra parte el cero, que es 0000 0000, tiene un 0 en su primer bit, por lo que el ordenador lo toma como un número positivo.

GEN MON EL DESENSAMBLADOR MAS ESTANDAR

***MSX DEVPAC* es el nombre conjunto de una** herramienta de trabajo **ensamblador/editor (*GEN*) desensamblador/monitor (*MON*) producido por *HI-SOFT* y comercializado en España por *SONY Indescomp*, muy parecido al estándar de *Zilog*, y es uno de los mejores, por no decir el mejor de los que hay actualmente en el mercado. En esta primera parte del artículo haremos un comentario general sobre los mismos. En el siguiente número trabajaremos con programas concretos que nos ejemplifiquen su utilidad.**

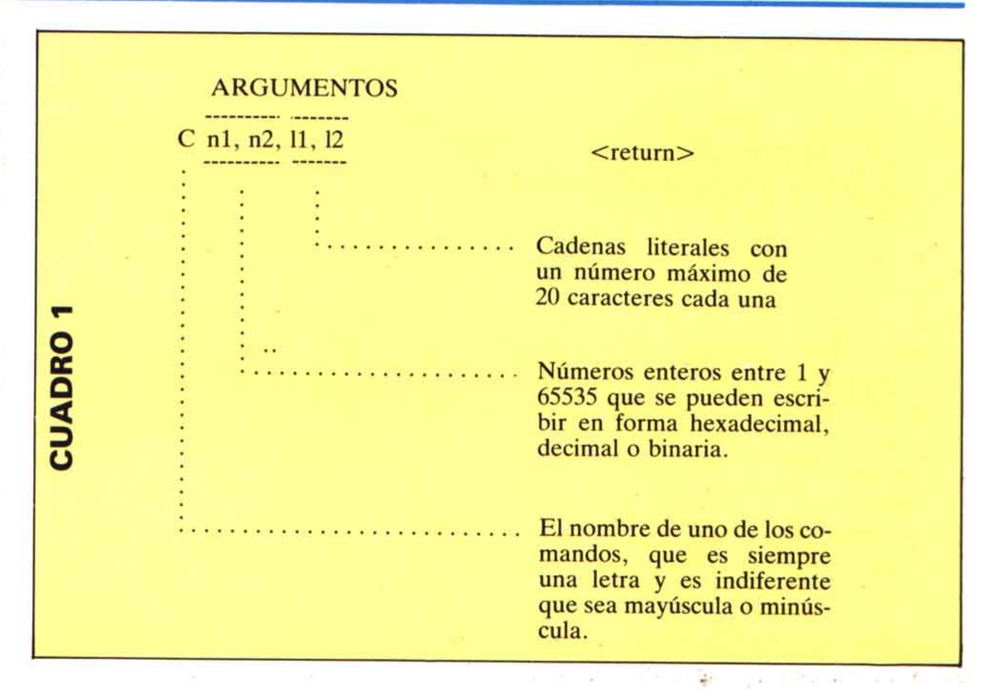
Pasemos a comentar de entrada el editor **GENMSX**. En primer lugar notamos que **GEN** se puede colocar en cualquier parte de la memoria del ordenador siempre que no se entre en contacto con el sistema operativo.

Así lo podemos ubicar de la siguiente forma:

```
BLOAD"CAS:GENMSX"
DEFUSR=&H88ØØ
A=USR(&HXXXX)
```

donde &HXXXX es la posición en la que se desea comience el editor.

Ejecutado el programa se entra directamente en modo de edición, mostrándonos la pantalla un amplio menú de elecciones. Podemos introducir entonces cualquiera de los comandos del editor, que se estructuran de la forma señalada en el cuadro 1:



Según los comandos, los argumentos pueden ser obligatorios o no, y algunos sólo tienen efecto si se declaran explícitamente determinados argumentos.

Veamos aquí los comandos, sintaxis y descripción:

* Comando B (Bye)
  Descripción: Devuelve el control al BASIC. Para volver a **GEN** desde este último hacer:

```
DEFUSR=&HFFFØ
A=USR(Ø)
```

* Comando *C* (*Current state*)
  Descripción: Informa sobre los valores actuales del delimitador y de los argumentos n1, n2, l1, l2. El delimitador generalmente es una coma.
* Comando *D* (*Delete*)
  Sintaxis: D<n1, n2>
  Descripción: Borrar del fichero de texto todas las líneas comprendidas entre la n1 y n2.
* Comando *E* (*Edit*)
  Sintaxis: E n
  Descripción: Edita la línea número n para permitir su modificación con ayuda del cursor y de los comandos de edición del MSX.
* Comando *F* (*Find*)
  Sintaxis: F n1, n2, l1, l2
  Descripción: Busca la cadena literal l1 a lo largo de las líneas comprendidas entre n1, n2.
* Comando *G* (*Get text*)
  Sintaxis: G, l1
  Descripción: Carga de cinta un fichero de texto.
* Comando *H* (*Help*)
  Descripción: Muestra la pantalla de ayuda.
* Comando *I* (*Insert*)
  Sintaxis: : I, n1, n2
  Descripción: Inserta texto, proporcionando automáticamente números de línea comenzando por la línea n1 e incrementándola a razón de la cantidad n2.
* Comando *J* (*Jump patch*)
  Descripción: Comando definible a voluntad del usuario.
* Comando *K* (*Konfigure*)
  Sintaxis: K n (n mayor o igual a 1ØØ bytes)
  Descripción: Reserva una zona de n bytes para almacenar las definiciones de las macroinstrucciones en

tanto dura el ensamblado.

* Comando *L* (*List*)
  Sintaxis: L n1, n2
  Descripción: Lista en pantalla las líneas n1 a n2.
* Comando *M* (*Move*)
  Sintaxis: M <n1, n2, n3>
  Descripción: Mueve un bloque de líneas que van de la n1 a n2, haciendo que se coloquen inmediatamente antes de la línea 3, para lo que se renumeran las líneas n3 y posteriores si es necesario.
* Comando *N* (*reNumber*)
  Sintaxis: N <n1, n2>
  Descripción: Renumera las líneas comenzando por n1 e incrementando un número n2 en el número de línea.
* Comando *O* (*Object*)
  Sintaxis: O, l1
  Descripción: Graba en cinta código objeto.
* Comando *Q* (*Q delim*)
  Sintaxis: Q, l1
  Descripción: Cambia el separador de los argumentos por el primer carácter de la cadena literal l1.
* Comando *R* (*Run*)
  Descripción: Ejecuta el programa objeto, siempre que no haya habido errores durante el ensamblado.
* Comando *S* (*Substitute*)
  Descripción: Sustitución de una cadena lineal.
* Comando *U* (*Upper line*)
  Descripción: Muestra el último número de línea.
* Comando *V* (*Verify*)
  Sintaxis: V, l1
  Descripción: Verifica un fichero de texto.
* Comando *W* (*Write memory*)
  Sintaxis: W n1, n2
  Descripción: Proporciona un listado en pantalla de contenido de memoria entre n1 y n2.
* Comando *X* (*teXt info*)
  Descripción: Muestra las posiciones en decimal del comienzo y final del fichero de texto.
* Comando *Y* (*Y print length*)
  Sintaxis: Y n1
  Descripción: n1 nos dice el tamaño de la página del listado por impresora.
* Comando *Z* (*Z print text*)
  Sintaxis: Z n1, n2
  Descripción: Listado por impresora.

Cuando se utiliza el comando *A* (*Assembly*) del editor, el programa pide:

TABLE SIZE: Cantidad de espacio para la tabla de símbolos y que ha de ser un número en decimal

OPTIONS: Las opciones de ensamblado, cuya respuesta debe ser un número decimal igual a la suma de las opciones que queramos y que a continuación se estipulan

opción 1: listado de tabla de símbolos al final de la segunda pasada de ensamblado
opción 2: No genera código objeto
opción 4: No realiza listado del ensamblado
opción 8: Listado por impresora
opción 16: Código objeto después de la tabla de símbolos
opción 32: Suprimir comprobación acerca de donde se sitúa el código objeto

GENMSX realiza el ensamblado en dos pasadas. Una halla los errores y compila la tabla de símbolos. Otra genera código objeto. Si se detecta un error el ensamblado se detiene y no se produce la segunda pasada.

El formato de cada sentencia en modo ensamblador (A) debe ser de la siguiente forma:

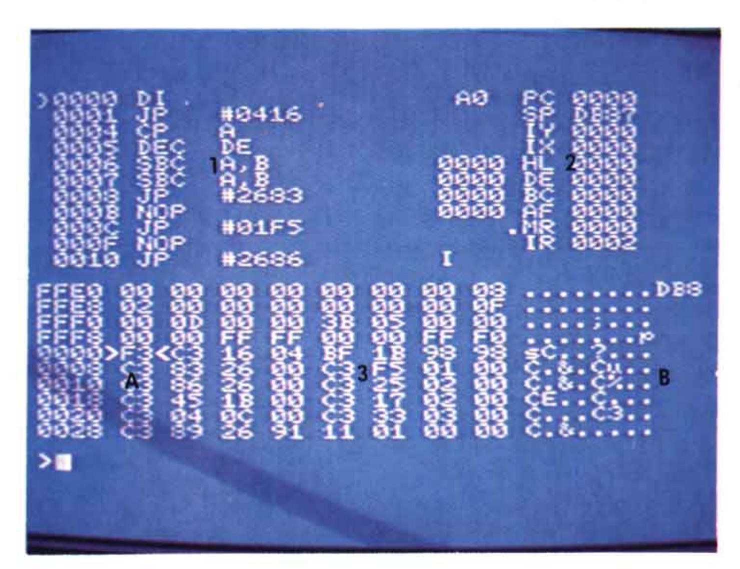
ETIQUETA
arranque:

CODIGO
LD

OPERANDO
HL, símbolo

COMENTARIO
; símbolo

Cada línea se procesa de la siguiente forma: Si el primer carácter de la línea es un punto y coma (;) la línea se considera un comentario y es ignorada por el ensamblador; si es un asterisco (*) se considera el carácter siguiente a dicho asterisco; si es *CR* (return) se ignora la línea. Pero si la línea comienza por un símbolo distinto a los tres an-



1. **Desensamblado de una porción de memoria que abarca II instrucciones.**
2. **El estado de registros.**
3. **Contenido de memoria en hexadecimal (A), y ASCII (B)**

teriores, el ensamblador recorre la línea en busca del símbolo (:); si lo encuentra todos los caracteres anteriores a éste serán interpretados como una etiqueta. Si el siguiente carácter no indica fin de línea o comienzo de un comentario (;), el ensamblador trata de interpretarlo como el comienzo de un cod-ope nemotécnico de un máximo de cuatro caracteres.

Otra cosa que favorece la utilización del GENMSX, es que además de los cod-opes del Z-80, éste reconoce y añade otros pseudonemónicos, tales como:

```
ORG,EQU,DEFB,DEFW,DEFS,
DEFM,ENT
```

o las pseudooperaciones condicionales:

```
IF,ELSE y END
```

Los comandos del ensamblador (no del editor) no actúan sobre el Z-80 ni sobre el código objeto, sólo modifican el listado del ensamblador. Un comando del ensamblador es una línea de fichero de texto, que comienza por el símbolo (*) y una letra mayúscula. Así existen los siguientes comandos: E, F, H, S, L, C, M. Este último *M lista (o no lista) las macroinstrucciones.

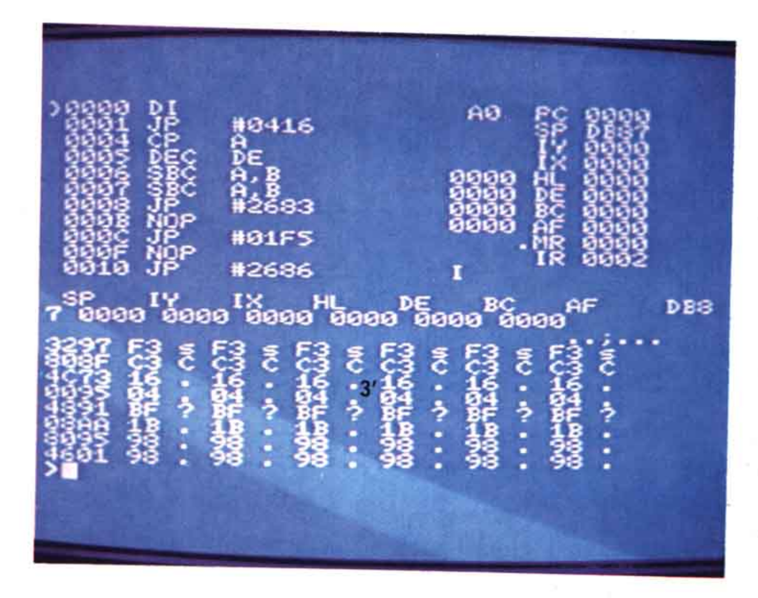
Las macroinstrucciones son otra de las ventajas de usar GENMSX. Una macroinstrucción es una rutina formada por la unión de operaciones elementales, pudiéndose utilizar hasta 32 argumentos, numerados de 0 a 31 llamados parámetros.

MON

MONMSX es un desensamblador/monitor que sirve para depurar programas en código máquina, convirtiendo la pantalla en un tablero de control lleno de informaciones que nos permiten visualizar el estado interno del ordenador.

MONMSX es reubicable, es decir, se puede cargar en cualquier parte de los 32K altos del mapa de memoria siempre que no entre en conflicto con el área de trabajo o la pila, es decir que lo podemos ubicar entre las direcciones &h8100 y &hD400.

Pantalla de MONMSX en comando 'S'



3'. estado de los registros en la parte 3'

Si lo queremos ubicar en otro sitio que no sea a partir de la &hB100, que es como viene si lo cargamos con 'LOAD"CAS:monmsx",r', hemos de teclear lo siguiente:

```
BLOAD"CAS:MONMSX",&HXXXX
```

donde &HXXXX es la diferencia entre la dirección a partir de la que se quiere ubicar (dirección nueva) y &h8800. Esto es:

```
&HXXXX=&H(direc.nueva)-
&h88ØØ
```

Para arrancar el programa se hace de la siguiente forma:

```
DEFUSR=&H(direc.nueva)
A=USR(&H(direc.nueva))
```

Este tablero de control puede imaginarse dividido en tres partes:

La primera, situada en la parte superior izquierda, visualiza el desensamblado de una porción de memoria que abarca once instrucciones.

La segunda, en la parte superior derecha, informa acerca del estado de los registros.

La tercera y última, que abarca toda la parte inferior de la pantalla, nos muestra el contenido de memoria de un sector de ochenta bytes centrado en la posición 0. Ésta, a su vez, se subdivide en contenido de la memoria en hexadecimal (subsector izquierdo) y contenido de la memoria en ASCII (subsector derecho).

A continuación veremos que **MONMSX** posee una gran diversidad de comandos, lo que lo hace uno de los mejores desensambladores/monitores de la actualidad.

Pasemos a comentar algunos de ellos:

- Comando *M*
  Sintaxis: M:XXXX
  Descripción: Selecciona el puntero de memoria. Ello significa que se debe introducir la dirección de memoria deseada (XXXX) cambiando el puntero a esa dirección.
- Comando *CURSOR DERECHA*
  Descripción: Incrementa en uno el puntero y el panel de memoria.
- Comando *CURSOR IZQUIERDA*
  Descripción: Decrementa en uno el puntero y el panel de memoria.
- Comando *CURSOR ABAJO*

Descripción: Incrementa en ocho el puntero y avanza una línea el panel de memoria.

* Comando *CURSOR ARRIBA*
  Descripción: Decrementa en ocho el puntero y retrocede una línea el panel de memoria.
* Comando .
  Descripción: Avanza el punto (.) de arriba abajo en el panel de estado de registros cambiando el registro seleccionado.
* Comando *R*
  Descripción: Copia el SP (puntero de pila) en el registro IR seleccionado en ese momento.
* Comando *CTRL + L*
  Descripción: Desensambla a partir del contador del programa (PC).
* Comando *J*
  Sintaxis: J: XXXX
  Descripción: Salto a la dirección XXXX del contador de programa (PC).
* Comando *Z*
  Descripción: Ejecuta cada vez una instrucción de los registros y actualiza la pantalla.
* Comando *I*
  Sintaxis: FIRST (direc. comienzo bloque)
  LAST (direc. final bloque)
  TO (direc. punto de destino)
  Descripción: Copia un bloque de memoria en otra parte de la memoria.
* Comando *P*
  Sintaxis: FIRST (direc. comienzo bloque)
  LAST (direc. final bloque)
  WITH (valor del byte)
  Descripción: Rellena un bloque de memoria con un byte determinado.
* Comando *G*
  Sintaxis: G: XX XX XX XX (en hexadecimal)
  G:" (cadena literal)
  G:% (código nemotécnico de una operación o de operandos)
  Descripción: Busca una sucesión de bytes en la memoria, a partir de la dirección que marca el puntero de memoria.
* Comando *N*
  Descripción: Busca de nuevo una sucesión de bytes que se han introducido con el comando G.
* Comando *S*
  Descripción: Cambia el tipo de panel de contenido de memoria.
* Comando *O*
  Descripción: Salto relativo del puntero de memoria.
* Comando *U*
  Descripción: Inverso del comando 'O'
* Comando *X*
  Descripción: Salto absoluto del puntero de memoria.
* Comando *V*
  Descripción: Inverso del comando 'X'
* Comando *CTRL + X*
  Descripción: Devuelve el control al BASIC. Si se desea volver a MONMSX:

```
DEFUSR=&HB8ØØ
A=USR(Ø)
```

Otros comandos del desensamblador son:

* Comando *CTRL + L*
  Descripción: Desensambla a partir del puntero de memoria.
* Comando *CTRL + N*
  Descripción: Desensambla la siguiente página de memoria mostrando las siguiente once instrucciones.
* Comando *CTRL + D*
  Descripción: Desensambla un bloque de memoria.
* Comando *CTRL + B*
  Descripción: Dispone una interrupción en la dirección del puntero de memoria.
* Comando *CTRL + R*
  Descripción: Suprime interrupciones.
* Comando *CTRL + K*
  Descripción: Modifica la configuración de los *slots*.

MATEMATICAS FACILES

Philips ha lanzado al mercado un lote de programas para sus nuevos ordenadores MSX2. Entre ellos figuran de gestión odontológica, PH-Stat2, bases de datos, procesadores de textos, juegos y el *PH-Math2*. A este último dedicamos, por su utilidad, este artículo.

El PH-Math2 está pensado para su uso por los estudiantes de BUP, COU y los universitarios de carreras técnicas. PH-Math2 se nos presenta como un lote de programas para aplicarlos a diferentes campos matemáticos. Está estructurado a partir de un menú principal a través del cual podemos acceder a los submenús de cada campo matemático determinado (funciones algebraicas, funciones trascendentes, operaciones con matrices...). Además, dispone de la posibilidad de acceder a una calculadora en cualquier momento, y así realizar las operaciones que creamos más oportunas para, después, retornar al punto de donde habíamos partido. Es decir, que si estamos operando con matrices y tenemos que hacer algún cálculo algo complicado, vamos a la calculadora, realizamos dicho cálculo, y volvemos al apartado de las matrices y podemos continuar sin necesidad de reempezar ni pasar por el menú principal.

Ahora vamos a ver con más detalle las posibilidades de este paquete matemático.

FUNCIONES ALGEBRAICAS

Este apartado trata las funciones polinómicas cuyos términos son del tipo a*X^n. En primer lugar, una vez hemos accedido a esta parte del programa, tenemos que entrar la función con la que vamos a trabajar. El grado de la función debe estar entre 2 y 19. Una vez introducida, el ordenador nos la mostrará en pantalla. Ahora ya podemos trabajar con ella y se nos presentan las posibilidades de: generación de funciones derivadas e integrales, resolución de ecuaciones, raíces reales, raíces racionales, raíces enteras, cálculo de máximos y mínimos, representación gráfica, análisis de función y curvas asociadas, cálculo de áreas y volúmenes relativos a la curva de la función y preparación de tratamientos estadísticos. Rápidamente comentaremos algunas opciones:

—Derivadas: genera la función derivada de la función polinómica con la que estábamos trabajando. Una vez



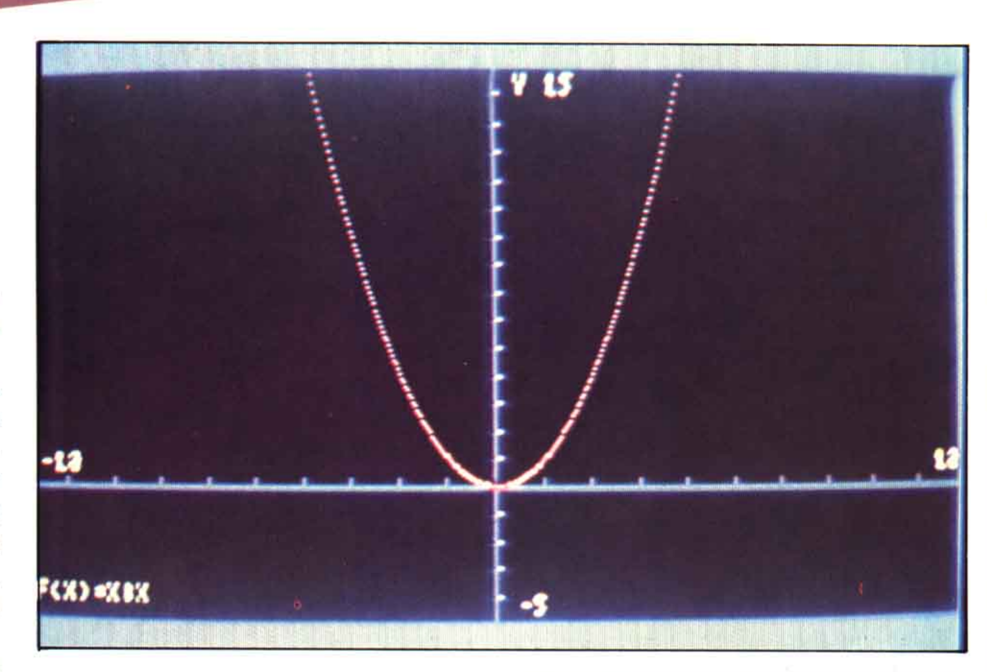
calculada la función derivada, sustituye a la original a todos los efectos de cálculo.

—Integrales: esta opción nos halla la primitiva de la función que nosotros introducimos. Es decir, si hacemos la derivada de ésta obtendremos la función que deseamos integrar. El término independiente adquiere el valor cero. Tras estas dos opciones podemos volver a la función original.

—Resolución de ecuaciones: halla todos los valores de x para los que la función toma valor cero. Como el ordenador resuelve la ecuación a partir del método de Newton, son necesarios dos parámetros: la aproximación, con la que deseamos obtener el resultado, y las iteraciones o los ciclos de proceso a utilizar para cada solución.

—Máximos y mínimos: el programa nos halla los puntos de la función donde se encuentra un máximo, un mínimo o un punto de inflexión.

—Estudio puntual de la función: dado un valor de la función, el ordenador nos da la imagen de este punto, la pendiente de la recta tangente en dicho punto. En caso de ser cero nos dice si se trata de un máximo o de un mínimo, si la función es cóncava o convexa en el punto a estudiar, además del término del polinomio que adquiere mayor valor absoluto en ese punto, y dicho valor.



FUNCIONES TRASCENDENTES

Igual que en el apartado anterior, lo primero que nos pide el ordenador es que entremos en memoria la función con la que queremos trabajar. Esta parte del programa ya entra en con-

Func. Alg.
Op. matrices
Nº Complejos
Usuario
Func. Trasc.
Res. Ecu. dif.
Borra arch.
Cod. Impresora
Fourier
Graf. Polares
Copia arch.
Fin
Calculadora
Graf. cartes.
Cat disco
CURSOR=INS ♦ MOVER= ‹ › ♦ ZOOM=DEL ♦ SALIDA=HOME ♦ COPIA=TAB
SALIDA DATOS
FUNCIONES
20
-20
20
-20
X= 0
Y= 1
Y'= -2
CONTROL
SISTEMA A LA ESPERA

flicto con las limitaciones del Basic, lenguaje en que está descrito en su mayor parte. Y esto lo limita y en definitiva nos limita a nosotros mismos. Las opciones de este apartado son:

—*Resolución de ecuaciones por el método de aproximaciones sucesivas:* para la resolución se nos piden tres parámetros. El primero de ellos es el valor de ensayo de la ecuación; el segundo, la aproximación con que deseamos el resultado; y el tercero, el mínimo de iteraciones a realizar. Una vez entrados todos estos datos el ordenador nos dirá para qué valores la ecuación es igual a cero.

—*Resolución de ecuaciones por tanteo:* los parámetros que se nos piden esta vez son el valor inferior del intervalo en el que se busca la raíz, el valor superior, la aproximación y el límite de situación. Para utilizar este método es condición necesaria que la función sea continua en el cuerpo real.

—*Máximos y mínimos:* busca los máximos y mínimos de la función a partir de los anteriores parámetros.

Hallar el valor de la función en un punto, el área entre abcisas y volumen o hacer la representación gráfica son las posibilidades que completan este apartado del programa.

CALCULO MATRICIAL Y SISTEMAS DE ECUACIONES

Esta parte del programa nos permite trabajar de forma fácil y eficaz con las matrices, y es una de las más útiles. En primer lugar, dimensionamos y ponemos en memoria una, dos o tres de las matrices que tenemos a nuestra disposición. Luego a partir de éstas podemos hacer las siguientes operaciones:

—asignación del valor de una matriz a otra;

—suma de dos matrices;

—diferencia de dos matrices;

—producto de dos matrices;

—cuadrado de una matriz;

—producto de una matriz por una constante;

—matriz inversa;

—matriz transpuesta;

—matriz adjunta;

—característica de una matriz;

—mayor y menor de los valores característicos de una matriz;

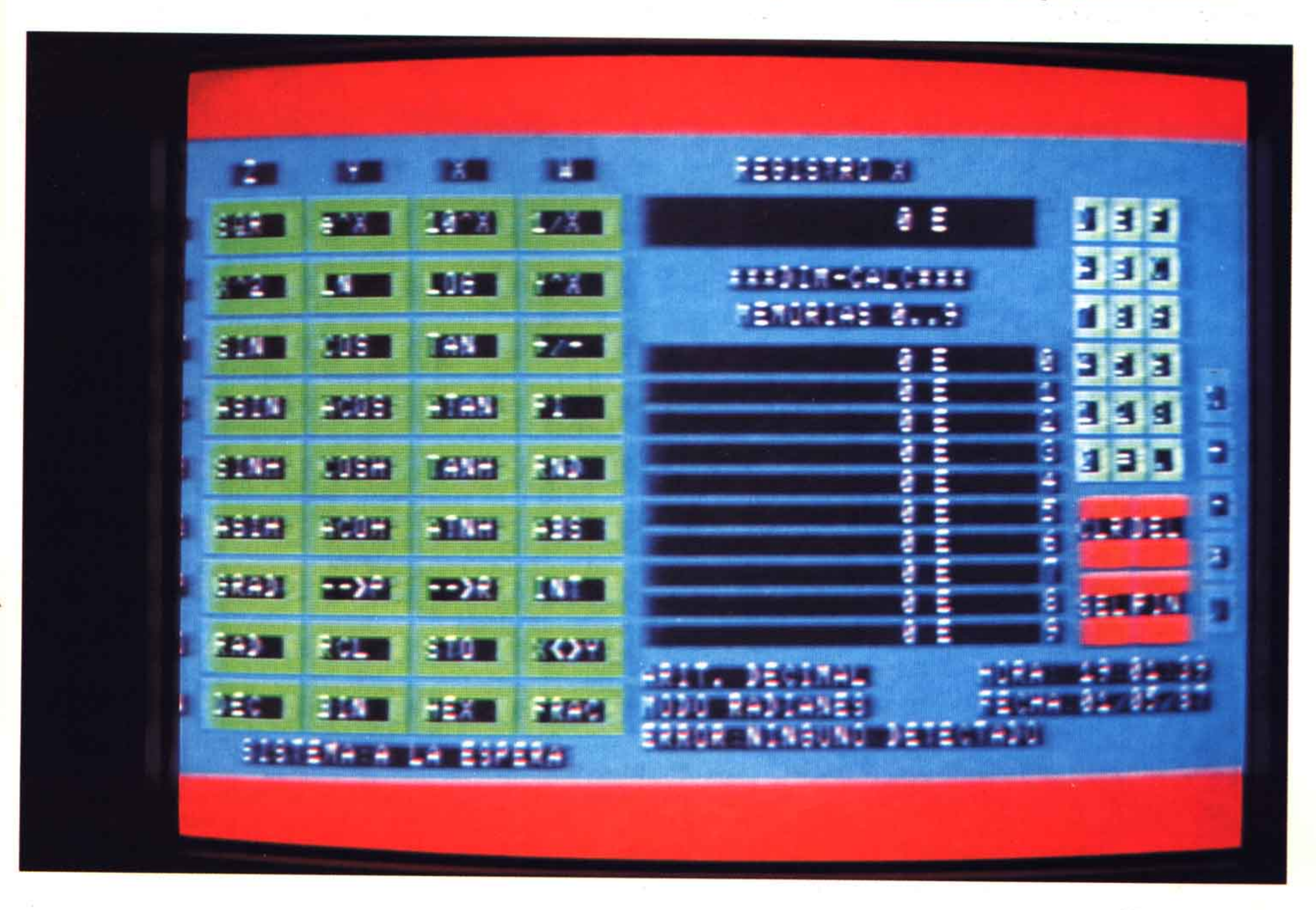
—determinante de una matriz.

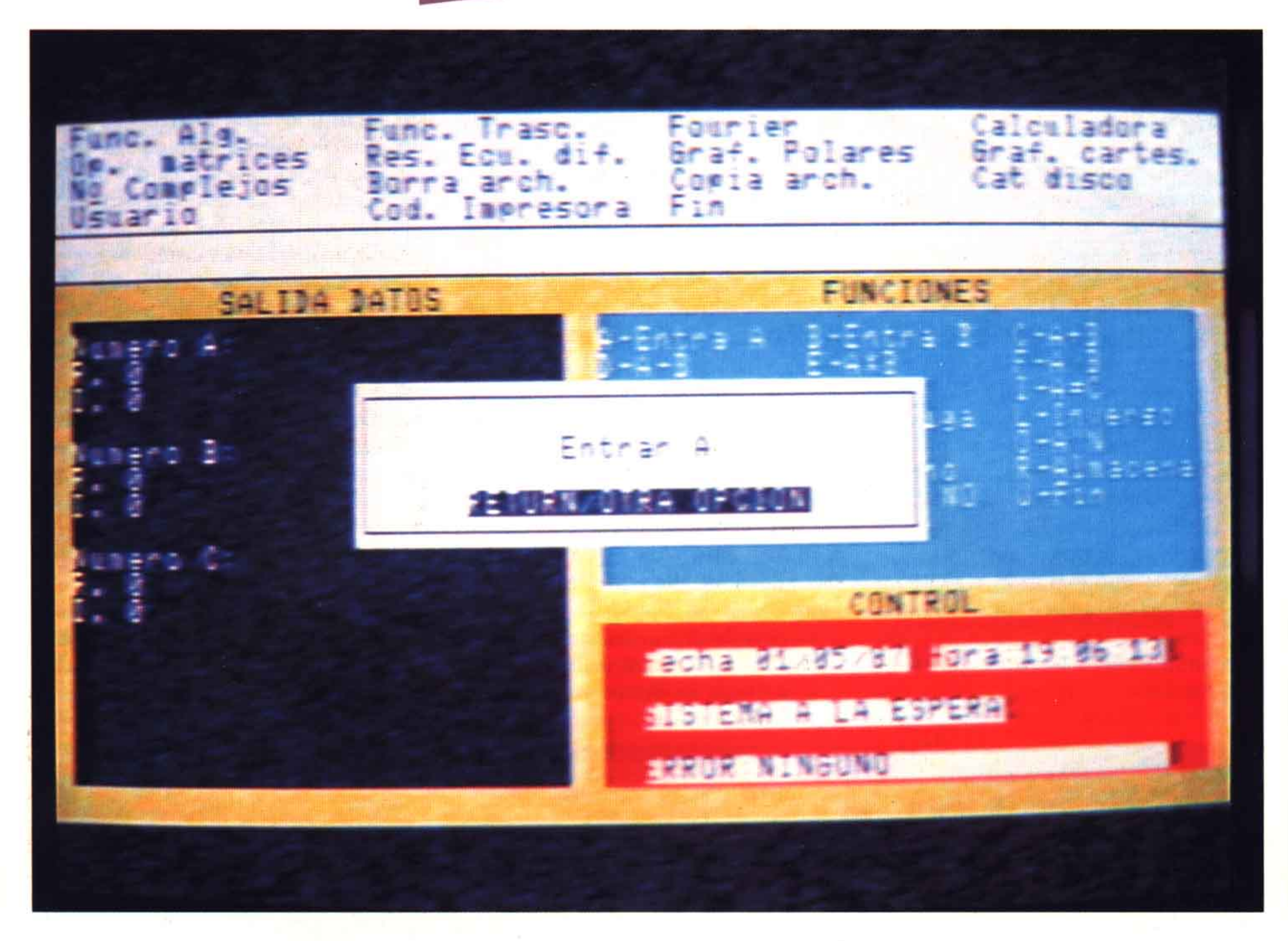
ECUACIONES DIFERENCIALES:

Este módulo nos permite resolver ecuaciones diferenciales a partir del método de Euler.

RESOLUCION SERIES FOURIER:

Este módulo, muy específico al igual que el anterior, tiene como única función el hallar los coeficientes de Fourier. Para ello tenemos que introducir la función directamente en el Basic del programa, entre las líneas 1ØØØØ y 11ØØØ y luego continuar con la ejecución de éste pulsando F1. Esto permite una mayor libertad y el poder introducir complicadas funciones.





CALCULADORA

Antes ya habíamos hablado de una calculadora, pero de lo que ahora vamos a tratar es de una calculadora científica, que se nos presenta como otra utilidad del programa y por esa razón es mucho más completa que la anterior, la cual jugaba un papel auxiliar.

La calculadora científica tiene 36 funciones específicas, más tres modos de operación aritmética (decimal, binaria y hexadecimal) y dos modos de operación angular (grados o radianes). Aparte de estas posibilidades hay un indicador de errores y un calendario.

NUMEROS COMPLEJOS

Este módulo trabaja con las variables A y B, que son designadas así en el menú y en la pantalla de resultados de las operaciones. Al entrar los valores de A y B, primero debemos indicar la parte real y luego la imaginaria.

Las operaciones que podemos realizar son: suma, resta, multiplicación y división. O bien hallar el inverso, el conjugado, el opuesto, las potencias de exponente entero, el exponencial de base e, el seno o el coseno, del número complejo.

Éstas son las opciones más importantes del programa, las que forman el cuerpo de éste, luego encontramos otras funciones auxiliares como catálogo del disco, código impresora y copia archivo disco. Además de la interesante opción que permite al usuario definir un programa y acceder a él a partir de PH-Math2, para, después de usarlo, volver al menú principal del paquete matemático.

Ya se nos advirtió, cuando salieron los ordenadores MSX, que su compatibilidad sería ascendente, es decir, que todo el software y el hardware de los primeros ordenadores serviría para los posteriores, pero no sería posible el caso contrario. Y este programa cumple con esta regla. En primer lugar, el formato con el que nos lo presentan es en disquet, así que si no se tiene una unidad lectora de disco ya no es posible contar con él. Y, en segundo lugar, todos los usuarios que dispongan de un MSX de primera generación y se hayan comprado una unidad lectora de disco tampoco podrán disponer de él, debido al mayor potencial del modo gráfico del MSX2 que, por ejemplo, nos permite trabajar con 80 columnas. Finalmente deseamos que este programa, realizado por DIMENSIONNEW, os ayude mucho en vuestros estudios.

COMPETICION EN EL BOSQUE

Si estos días vais por el bosque oiréis un gran revuelo, porque los animales están en plena competición. Los pingüinos han inventado un nuevo juego, y ahora ya se celebran los octavos de final. Los dieciséis supervivientes a las duras pruebas de clasificación aspiran a salir victoriosos. Nos hemos infiltrado para ver uno de estos partidos y verdaderamente éste es un juego original y emocionante. En un principio no parece gran cosa, pero a medida que pasan los minutos su atractivo crece. Y si te atreves a jugar, verás que desde dentro es todavía más emocionante.



Al inicio se colocan cinco bolas en dos extremos opuestos de una mesa. Los jugadores se sitúan tras ellas, sobre una tarima y empiezan a lanzárselas. Ganará el participante que logre colocar todas las bolas en el campo de su adversario. Para lograr este objetivo todo está permitido, hasta derribar al contrario lanzando las bolas sobre él. Y, de hecho, éste es el único sistema para ganar. El público enfervorecido jalea a los participantes y los anima desde las gradas, que están llenas a rebosar. Entre partida y partida el vencedor juega unas pantallas de bonificación, en las cuales sólo se aumenta la puntuación de dicho jugador. Son como una recompensa al esfuerzo realizado.
Los partidos se juegan al mejor de tres sets. Un set se acaba cuando todas las bolas se encuentran en uno de los extremos de la mesa. Pero si antes de que esto ocurra se agota el tiempo, ganará quien menos bolas tenga en su campo. En caso de empate se jugará un período extra con once bolas en lugar de diez.

DATOS GENERALES

TITULO: Penguin Wars

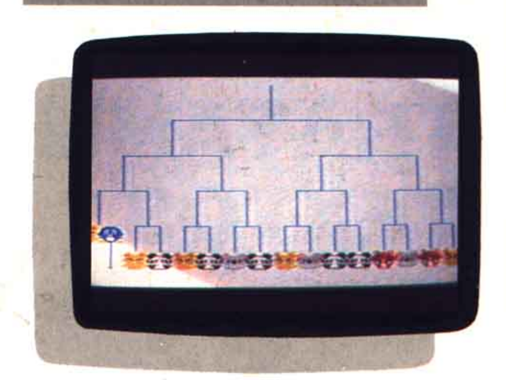
FABRICANTE: Ascii

CLASE DE PROGRAMA: Juego

FORMATO: Cartucho ROM

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD:	9
INTERES:	8
GRAFICOS:	8
COLOR:	7
SONIDO:	8
TOTAL:	40



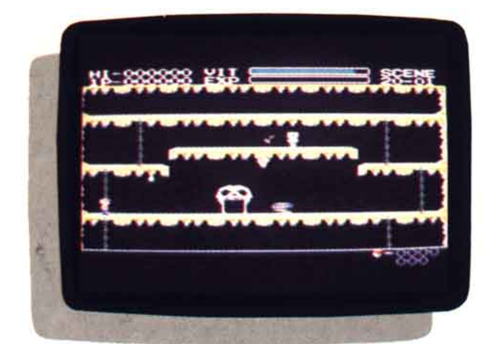
★★★★★★★★★★★★★★★★★★★★★★★★★

LA PELICULA CONTINUA

Una pandilla de chicos, llamados los **Goonies**, han encontrado el mapa de un tesoro en el ático de la casa de Mickey. Todos ellos, **Mickey, Brand, Mouth, Chunk** y **Data**, más dos muchachas que se unen al grupo, **Andy** y **Stef**, deciden ir a buscar el tesoro de **Willy el Tuerto**, un famoso pirata de la región. Según el mapa, el tesoro se encuentra en la localidad costera de Cauldron Point, un lugar lleno de túneles y pasadizos secretos. Allí se esconde la banda de los **Fratelli**, que secuestra a la banda de los **Goonies** y los tienen prisioneros. Pero nadie parecía contar con **Sloth**. Este personaje, protagonista del juego, arriesgando su propia vida, se ha aventurado a rescatar a los **Goonies** y a encontrar el tesoro. Y para conseguirlo tendrá que luchar contra la banda de los Fratelli y con los peligros de las cuevas de los **Muelles de Goon**. La primera misión de **Sloth** es liberar a los **Goonies** y sacarlos de sus celdas. Para ello



tendrá que recorrer todas y cada una de las pantallas de los cinco niveles de que consta el juego, encontrar las llaves y abrir las diferentes celdas en que pueden hallarse encerrados sus amigos. Una vez conseguido todo esto, deberá hallar la puerta que les





DATOS GENERALES	
TITULO: Los Goonies	
FABRICANTE: Konami	
CLASE DE PROGRAMA: Juego	
FORMATO: Cartucho ROM	

CALIFICACION (Sobre 10 ptos.)	
ORIGINALIDAD:	8
INTERES:	8
GRAFICOS:	8
COLOR:	9
SONIDO:	8
TOTAL:	41



conducirá hasta el próximo nivel. Allí, el calvario empezará de nuevo. Pero no os preocupéis, al final de cada nivel se nos revela una contraseña, que podemos utilizar cada vez que reanudamos el juego, para hacerlo a partir del último nivel al que llegamos.
A partir de entonces el desenlace de la película dependerá de vosotros.

★★★★★★★★★★★★★★★★★★★★★★★★★

TRATAMIENTO DE TEXTOS

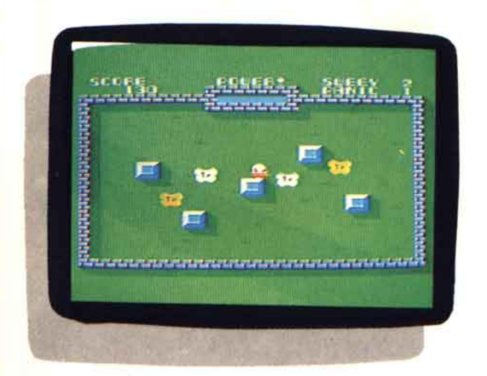
Idealogic nos presenta un procesador de textos sencillo pero eficaz. Tiene como mejor arma el fácil manejo y la presentación en 29 columnas, cosa que da un gran tamaño a las letras, lo que nos permite leer en pantalla, resaltándonos aún más los errores que podamos haber introducido en el texto.
En la parte inferior de la pantalla visualizamos toda la información necesaria para trabajar con el procesador. Es decir, línea y columna en que está el cursor, memoria libre que nos queda y los títulos de los cinco menús del programa. Estos menús son de acceso directo y sólo se ha de pulsar una de las cinco teclas de función para visualizarlos en pantalla. Con F1 accedemos al menú que nos permite trabajar con bloques de texto, y consta de ocho interesantes posibilidades: inicio de bloque, final de bloque, borrar, copiar, grabar, insertar, mover o limpiar el bloque señalado. Gracias a F2 visualizamos el menú de disco o de cinta, según nos convenga. Básicamente nos permite almacenar y recuperar textos. F3 llama a pantalla al menú de búsqueda, que nos permite escoger entre una de las siguientes opciones: poner etiquetas, salto de etiquetas, buscar una palabra o sustituir una palabra. Para acceder al menú de controles de impresora debemos pulsar F4, y con F5 vamos directamente al menú llamado: visualizar. Éste tiene como primera opción el paso del texto a la impresora, y como segunda la visualización de cómo quedaría el texto en un folio. Ésta viene a suplir los inconvenientes que tiene trabajar en 29 columnas.



DATOS GENERALES	
TITULO: Ideatext	
FABRICANTE: Idealogic	
CLASE DE PROGRAMA: Procesador de textos	
FORMATO: Cartucho ROM	

CALIFICACION (Sobre 10 ptos.)	
ORIGINALIDAD:	8
INTERES:	9
GRAFICOS:	8
COLOR:	6
SONIDO:	6
TOTAL:	37

LA DULCE BELLOTA



Una bellotita se ha perdido en un jardín encantado y nosotros debemos ayudarla a encontrar el camino de salida. El jardín está lleno de desconocidas y peligrosas criaturas: los **Peepas**, que no son más que judías y pimientos que intenta comérsela. Estos enemigos cambian de actitud según sea su color. Éstos intentarán atacar a nuestra pobre bellota, que sólo puede defenderse lanzándoles unas piedras que lleva en los bolsillos. La bellota controla la fuerza con que lanza las piedras, pero lo que no sabe es que éstas irán rebotando por toda la pantalla hasta agotar su energía, alcanzar a un **Peepa** o a ella misma, de modo que se habrá producido un suicidio. Además, en la pantalla encontramos los **Bop-Bumpers**. Unos cuadrados que siempre permanecen quietos, pero igualmente las piedras rebotan en ellos y al hacerlo los **Peepas** se vuelven más peligrosos y nos lo demuestran cambiando de color, pero ahora al matarlos lograremos una mayor puntuación. Al acabar con todos los **Peepas** de una pantalla pasamos a la siguiente y así sucesivamente hasta lograr llevar a la pequeña bellota de nuevo a casa.

DATOS GENERALES

TITULO: Sweet Acorn

FABRICANTE: Electric Software

CLASE DE PROGRAMA: Juego

FORMATO: Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD:	8
INTERES:	6
GRAFICOS:	7
COLOR:	7
SONIDO:	6
TOTAL:	34



****

DISPARA, DISPARA, DISPARA

Knightmare es uno de esos juegos que te sorprenden por su calidad, una calidad a la que ya nos tiene acostumbrados Konami, pero una vez más les debemos felicitar. El programa es un juego de arcade, pero hecho con mucha imaginación y buen gusto. Para poneros un ejemplo os diremos que el juego tiene una tecla de pausa, y al activarla el guerrero protagonista se pone a dormir bien protegido por una manta, y hasta parece roncar. Y este realismo se encuentra en todo el juego. Los gráficos son originales y su diversidad es tal que resulta imposible describir a todos los enemigos que encontramos a través de los ocho niveles del juego. Pero es necesario destacar a unos en concreto, los cuales llegaréis a odiar: son los que encontramos al final de cada nivel y sólo si logramos matarlos pasaremos de pantalla. La parte sonora del programa tampoco desentona con la alta calidad del



DATOS GENERALES

TITULO: Knightmare

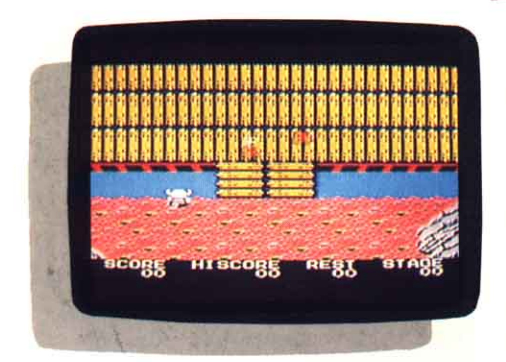
FABRICANTE: Konami

CLASE DE PROGRAMA: Juego

FORMATO: Cartucho ROM

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD:	9
INTERES:	9
GRAFICOS:	9
COLOR:	8
SONIDO:	8
TOTAL:	43





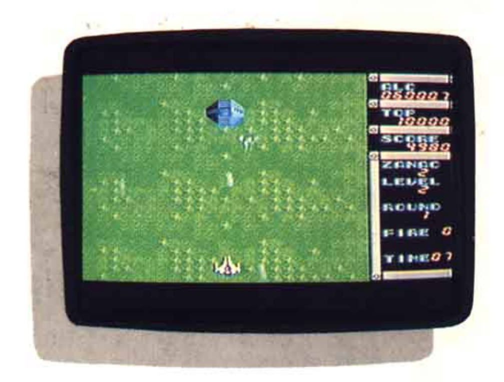
juego en general. En todo momento hay una música de fondo y sobre ésta se oyen los sonidos que se producen al hacer cosas específicas, como son: disparar, matar o conseguir una vida extra. Los componentes adicionales del programa son diversos, y ahora resaltaremos algunos. Cambio de arma: una bola mágica nos permite seleccionar nuestra arma preferida, dentro de la amplia gama que se nos ofrece. Los pasadizos secretos: a lo largo de nuestro camino hay escondidos unos pasadizos secretos y si los descubrimos tenemos la posibilidad de pasar directamente de nivel, esquivando un montón de dificultades. Los puentes secretos: en los niveles superiores antes de cruzar un río debemos encontrar dónde se hallan los puentes, de lo contrario acabaremos ahogados en el agua.





★★★★★★★★★★★★★★★★★★★★★★★★★★★

LA BATALLA GALACTICA



El programa que comentamos a continuación no se encuentra en el mercado español, pero creemos que es interesante ir informando de cuáles son los programas que van surgiendo allende nuestras fronteras. Ya que si tenéis la oportunidad de acceder a estos mercados sabréis qué programas os pueden resultar más interesantes.
El programa en cuestión se llama **Zanac**. Es un juego de arcade y más concretamente un «masacramarcianos». Pero lo que le diferencia de otros muchos programas de esta misma clase es su calidad. Si alguna cosa en concreto podemos destacar es el punto justo en que están escogidos los elementos que lo componen, ni le faltan ni le sobran cosas.
Al empezar la partida vemos una pequeña nave en la parte inferior de la pantalla, la cual posee dos potentes disparos. El primero es un eficaz láser y el segundo es un disparo adicional, el cual podemos ir cambiando a lo largo de la partida.
El nivel al que jugamos lo podemos escoger nosotros mismos durante la partida. En un principio jugamos al nivel 0 y el láser lanza un único disparo, pero si lo deseamos y somos suficientemente hábiles podemos ir aumentando el número de disparos de nuestro láser hasta lograr el máximo que son tres. Imaginaros un arma triple más un disparo adicional, es lógico pensar que lo arrasaremos todo.
Pero no es así, porque ahora estaremos en el sexto nivel y los enemigos no nos dejarán ni respirar. Además, el juego consta de rounds, y en cada round cambian los enemigos y se vuelven más peligrosos y agobiantes. La única salida es elevar nuestra habilidad.

DATOS GENERALES

TITULO: Zanac

FABRICANTE: PONYCA

CLASE DE PROGRAMA: Juego

FORMATO: Cartucho ROM

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD:	8
INTERES:	9
GRAFICOS:	9
COLOR:	9
SONIDO:	9
TOTAL:	44





AL SERVICIO DE LA RAF

Seguro que ya conocéis este programa de la **U.S. Gold** o habéis oído hablar de él. Y es que su fama no es fruto de la casualidad. **Dambusters** es un juego de estrategia y un simulador de vuelo a la vez, que se basa en una legendaria misión que llevó a cabo el **Escuadrón 617** durante la Segunda Guerra mundial, sobre un objetivo alemán. Debido a la gran dificultad del juego, se nos permite empezar desde tres niveles diferentes. El nivel más sencillo, «Practicar aproximación a la presa», nos acerca al objetivo sin preocuparnos de los **ME110**, de los reflectores ni de los globos. El segundo nivel de dificultad, «Opción teniente de vuelo», nos sitúa ya en el aire, encima del canal de la Mancha. Pero esta vez ya no estaremos solos: los enemigos estarán persiguiéndonos y nos harán la vida imposible. El tercer y último nivel, «Opción jefe de escuadrilla», ya no nos da ninguna facilidad. Todo depende de nosotros, despegar, seguir el mejor rumbo posible, despistar a los enemigos y llegar a destruir nuestro objetivo. Os aseguro que esto no es nada fácil, ya que solos tenemos que hacer el trabajo de toda la tripulación. Presionando las teclas numéricas del 1 al 8 accedemos a las siguientes pantallas: 1. piloto; 2. artillero; 3. artillero trasero; 4. bombardero; 5. navegador; 6. pantalla del primer ingeniero; 7. pantalla del segundo ingeniero, y 8. informe de situaciones y daños. Cuando una de estas posiciones se encuentra en dificultades parpadea en la parte inferior de la pantalla el número correspondiente.

DATOS GENERALES

TITULO: Dambusters

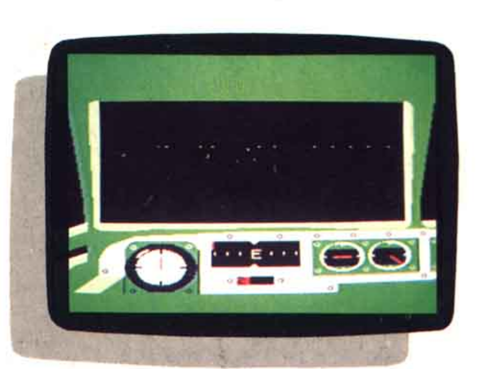
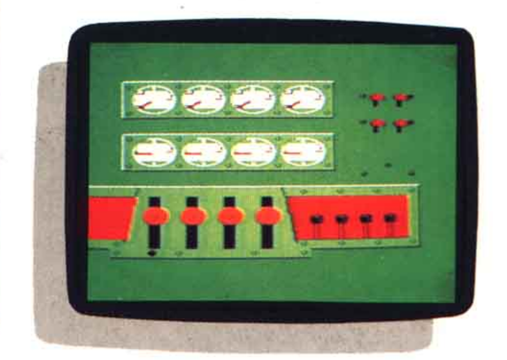
FABRICANTE: U.S. Gold

CLASE DE PROGRAMA: Juego

FORMATO: Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD:	9
INTERES:	9
GRAFICOS:	7
COLOR:	8
SONIDO:	7
TOTAL:	40

FINTA, SALTA, Y... CANASTA

Dinamic, basándose en el jugador de la NBA, **Fernando Martín**, ha creado un juego relacionado con el deporte del básquet. El juego sigue las reglas y las normas del básquet, pero sólo se enfrentan dos jugadores. Tenemos dos opciones, jugar contra el ordenador o hacerlo un jugador contra otro. En ninguno de los dos





casos existe la opción de seleccionar un nivel de dificultad.

El partido se desarrolla en un campo que reúne todas las condiciones necesarias, y un numeroso público aplaude nuestras jugadas. Si observamos a **Fernando** aprenderemos a realizar todos los movimientos que él ejecuta con una perfección envidiable. Bota con la mano derecha, con la izquierda, finta, realiza giros en el aire y encesta gracias a un gancho o machacando el balón en el aro. Debido a que jugamos en la NBA seguimos la norma que allí rige respecto al número de personales, es decir, se nos expulsa al cometer la sexta personal y no la quinta como sucede en la FIBA. Además de las faltas personales se nos penalizan otras infracciones, como los pasos, los dobles o el campo atrás. Además del juego destaca la presentación que le precede. En ella vemos una caricatura de **Fernando Martín** que mientras nos sonríe hace malabarismos con un balón. Si os animáis a jugar quizás os vea **Díaz Miguel** y os coloque en la selección.

DATOS GENERALES

TITULO: Fernando Martín

FABRICANTE: Dinamic

CLASE DE PROGRAMA: Juego

FORMATO: Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD:	8
INTERES:	8
GRAFICOS:	7
COLOR:	6
SONIDO:	6
TOTAL:	35

★★★★★★★★★★★★★★★★★★★★★★★★★

MISION SUICIDA

El servicio de inteligencia de nuestro ejército nos ha seleccionado para llevar a cabo una dura y difícil misión. Nosotros, orgullosos de haber sido seleccionados como el soldado idóneo para una misión de tanta importancia, hemos aceptado rápidamente. Después de unas duras semanas de entrenamiento, y sin saber aún el objetivo de la misión, ni el plan a seguir, nos han comunicado que mañana volaremos hasta el objetivo. A la mañana siguiente nos proporcionan una ametralladora y cinco granadas. Al alcanzar el campo enemigo nos hacen saltar del helicóptero y nos mandan destruir al máximo esas instalaciones y liberar a los compañeros que están allí presos. Hemos aceptado una misión suicida y ahora ya no podemos renunciar a ella, aunque esa idea en ningún momento ha pasado por nuestra cabeza. Ahora sólo dependemos de nosotros mismos y de una vieja ametralladora. Los enemigos, desorientados, caen como moscas. Somos como un huracán, nada se resiste a nuestro ataque. A medida





DATOS GENERALES

TITULO: Who Dares Wins II

FABRICANTE: Alligata

CLASE DE PROGRAMA: Juego

FORMATO: Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD:	6
INTERES:	7
GRAFICOS:	7
COLOR:	6
SONIDO:	7
TOTAL:	33



que avanzamos en el campamento enemigo encontramos coches, helicópteros, casetas y fortines, cosas que tenemos que destruir. Además de nuestros compañeros, a punto de ser ejecutados, o de granadas de repuesto. En nuestra mente sólo existe una idea: avanzar y destruir. Pero todo tiene su fin, y bien por cansancio o por mala fortuna una bala nos alcanza y caemos allí mismo, pero sabiendo que hemos cumplido con nuestra misión. Esta historia ha sido creada por Alligata en una versión para MSX de este tipo de juegos. Prepárate, tu vida dependerá de ello.



★★★★★★★★★★★★★★★★★★★★★★★★★

EL ROMPECABEZAS





DATOS GENERALES

TITULO: Confused

FABRICANTE: BYTE BUSTERS

CLASE DE PROGRAMA: Rompecabezas

FORMATO: Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD:	9
INTERES:	8
GRAFICOS:	7
COLOR:	7
SONIDO:	6
TOTAL:	37

Confused es un rompecabezas llevado a cabo de forma muy original, gracias a que aprovecha las ventajas de estar hecho por ordenador. En un primer menú podemos escoger entre diez figuras diferentes, y si queremos las visualizaremos en pantalla. Éstas las podemos dividir desde en cuatro hasta noventa y seis cuadrados. Después de haber pasado por el menú principal y haber hecho la selección de todos los datos, podemos empezar la partida. No se trata de un rompecabezas convencional, las fichas no tienen una imagen asignada, sino que por ellas corren las imágenes, y tenemos que ponerlas de tal manera que la imagen siga su curso en el dibujo y no quede partida o aparezca por el otro extremo, formando una figura incoherente. Para llegar a formar el rompecabezas tenemos unos comandos de ayuda: visualización de la imagen, inversión de los cuadrados o paralización de la imagen. Además de una opción de ayuda, que coloca los cuadrados que escojamos en su debido lugar. Si nos rendimos podemos volver directamente al menú principal.

Mediante una flecha, dirigida por los cursores, podemos acceder a todos los controles, sin necesidad de tener que tocar ninguna de las otras teclas. Esto da una gran agilidad al programa y nos facilita su manejo. Si disponéis de tiempo y tenéis ganas de volveros locos delante del ordenador, este programa os satisfará.





EL ZOCO



Vendo/cambio juegos MSX; poseo Knight Lore, Jack the Nipper, Gunfright y otros (15 más).
José Luis Canales
La Concha de Villaescusa (Cantabria).

Vendo Philips VG-801o MSX con unos 40 programas (ALIEN-8, JACK THE NIPPER, BATMAN, TENNIS, PROFANATION, KNIGHTMARE, CAMELOT WARRIORS, GOLF, THE WAY OF THE TIGER, etc...), Interface Centronics, Ampliación 64K, cassette, cables, manuales. Todo por 22.000 ptas. Llamar a: Javier Torres Tl: (91) 204 43 15. C./ Virgen de la Oliva n.° 42, 28037 Madrid.

Vendo Cartucho de Expansión de Memoria de l6K (SONY) por 4.000. Dirigirse a: Mercedes Muñoz, C./ Teruel n.° 9, 3.° 6.ª Tl: 363 93 00 (horas de oficina) 46008-Valencia.

Intercambio programas, trucos e ideas para el sistema MSX. Dirigiros a: Diego Sánchez Alonso, C./ Montoya n.° 1, 7.° B Alcantarilla (Murcia) 30820.

Intercambio programas MSX en disco de 3 ½ pulgadas. Irene Juardos Revilla, C./ Garita n.° 19;07015-Palma de Mallorca. Tl: (971) 40 36 59.

Intercambio programas, trucos e ideas con usuarios de MSX. Poseo todos los de Kowami, Gremlim Graphics, Dinamic, Ultimate, etc. Enviar lista. Escribir a: Pablo Fernández Mastache, C./ Gran Vía n.° 104, 3.° B.36203 Vigo (Pontevedra).

Compro cartucho de ampliación de memoria de 16K (que no supere las 6.000 ptas.) o bien de 64K (que no supere las 15.000 ptas.) Manuel García Solano, C./ Ballesteros Villanueva n.° 1, Orihuela (Alicante).

Vendo ordenador SONY HIT BIT 75P, 64K (teclado profesional, manuales y cables). Poco uso, comprado hace un año. Precio 43.000. Además regalo base de datos (DATABA) y procesador de textos (COMPOR) de Philips. Luis Miguel Sastre, C./ Espoz y Mina n.° 15, 3.° D-31200-Estella (NAVARRA).

Compraría ampliador de memoria para HIT BIT SONY MSX de 64K y también cartuchos de juegos.
Iván Regalado, C./ Concepción Arenal n.° 136 entlo 1.° 08027 Barcelona. Tl: 340 01 41.

Cambio 20 títulos primeras marcas (Knight, Alien, Gunfright, Hyperl y II, Boxing, Yie, Biliards...) por uno de los siguientes juegos: (Green Beret, Night Mare, The Goonies, Yie II, Tennis Konami, Golf Konami, Grog's Revenge (originales o no) y además VENDO Antartic Adventure, Track & Field II, Senjyo, Chiller, Finders Keepers (todos originales) por 9.500 ptas. También los vendo por separado. Llamar o escribir a: Rafael López Fernández, C./ Indíbil y Mandonio n.° 34, 3.° 2.ª 25003 Lérida. Tl: (973) 26 26 54.

Vendo, compro e intercambio programas MSX, MSX-DOS y CP/M en disco 3.5", estoy interesado especialmente en aplicaciones. Poseo WORDSTAR, L80, M80, SUPERCALC, etc. Todos excelentes programas CP/M, y más de 200 programas MSX comerciales. Oscar Vallejo Marcelo. Avd. Madrid n.° 9, 3.° 3.ª esc. C. 08028 Barcelona.

Vendo ordenador MSX SONY HB 55P por 35.000 ptas. Incluyo ampliación de memoria 16K y cassette, así como cartucho de juego Tennis y otros en cinta, Interesados llamar al tl: (93) 211 42 24 de Barcelona.

Intercambio toda clase de programas (más de 200) en código máquina. Escribir a: Miguel Jiménez Pizarro, C./ Morenito de Algeciras n.° 4, 4.° C. Algeciras (Cádiz). Tl: (956) 60 45 86.

Vendo ordenador MSX Panasonic CF2700,80 Kram, espacio para dos cartuchos (comprado hace dos meses), manuales y revistas sólo por 27.950 pesetas.
Llamar al teléfono (93) 331 99 02 Barcelona.

Cambio monitor DYNADATA fósforo verde, con contraste, brillo y volumen (en perfecto estado) y algo de dinero, por monitor en color.
Escribir a: Jesús Mingorance Aguilera, C./ Mulhacén n.° 9-11 18005 Granada. Tl: 25 20 03.

Intercambio juegos de código máquina (Goonies, BatMan, etc...), casi todos los de Konami. Ignacio Aguilera Espejo, C./ Sevilla n.° 16. Mairena del Aljarafe (Sevilla).

Intercambio programas MSX, tengo los mejores títulos. Konami, Erbe, Activision... Juan Carlos Patricio Asensio, C./ Rodrigo Pertegás, n.° 48, 11.°. 46023 Valencia.

Vendo juegos MSX entre ellos (Knight Lore, Boxing, Camelot Warriors, Nightmare...etc) a precio muy barato. Envíos contra reembolso. Escribir a: Mari Carmen Alonso Tomás, C./ San José de la Vega n.° 9, 9.° 46011 Valencia.

Intercambio bibliografía y SOFTWARE de gestión en cinta para MSX o Philips P200. Sebastián Briones Moyano C./ Pintor Carbo n.° 2, P.° 5, P.ª 10. Castellón 12003. Tl: (964) 22 33 23.

Intercambio programas MSX tanto en cinta como en disco 3.5" sin interés económico. Enviar relación. Luis Amado Rego, C./ Puente, 25-3. 36500-Lalin. Pontevedra.

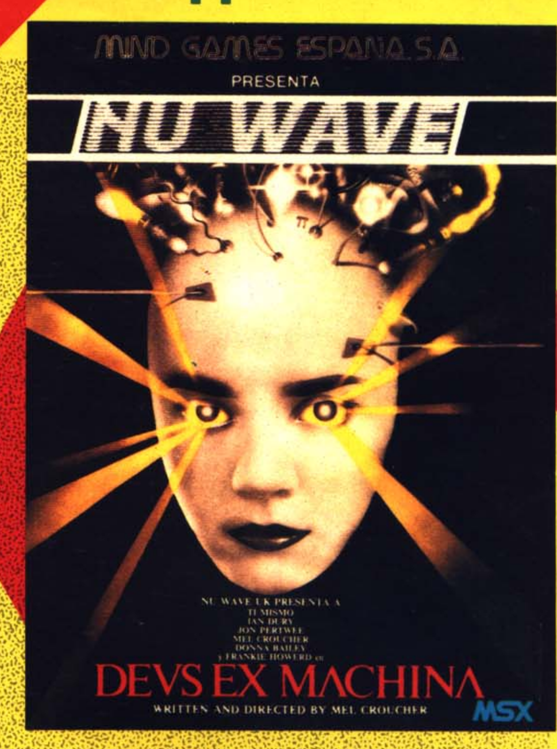
Vendo ordenador MSX X'PRESS, unidad de discos incorporada, de 3.5", más de 40 juegos, Joystick, sistemas operativos CP/M y MSX-DOS, cassette, manuales, libros y revistas, todo por 67.000 ptas.
Tl: 311 44 14. Preguntar por Gabi, de 19 a 22 h.

Atención Zx Spectrum Plus, con todos sus accesorios y manuales, más interface Kempston y joystick Quick Shot II, todo en perfecto estado y con embalajes originales; revistas Microhobby (colección completa) con curso de Basic y C/M, juegos originales de calidad y curso de Basic de Investrónica. *Todo* a un precio estupendo. Llama al teléfono (976) 51 07 15, Zaragoza.

NO OLVIDES EL TELEFONO... ☎

Cuando, por cualquier motivo, nos escribas, no olvides indicar tu número de teléfono. Así nos será más fácil y rápido ponernos en contacto contigo. Gracias.

¡¡LOS MEGA HITS MSX !!

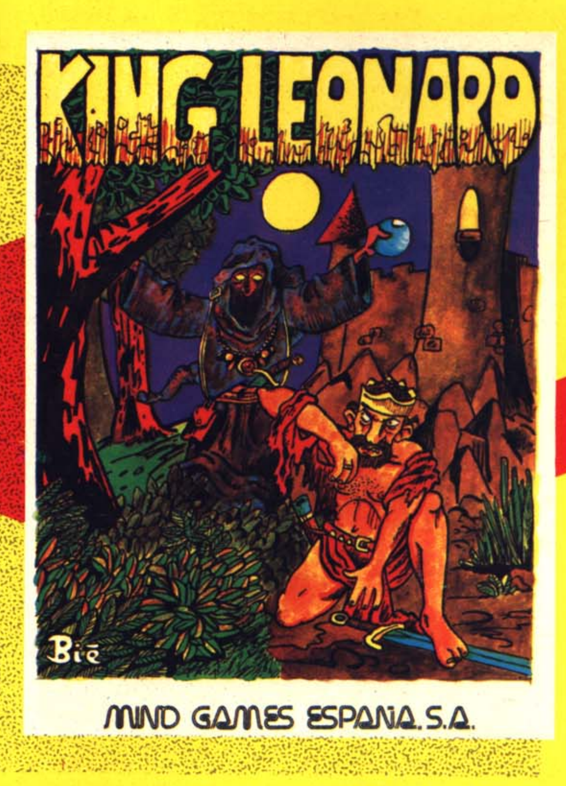


¡LA AUDIO-VIDEO AVENTURA MAS ALUCINANTE JAMAS CREADA...!



EL TUNEL DEL TIEMPO
Es un ameno juego educativo que nos proporciona la oportunidad de viajar a través del tiempo y del espacio, recorriendo las tierras de nuestro planeta y contactando con los personajes y acontecimientos más relevantes de las distintas eras históricas.

Disponibles en:
CASSETTE
y
DISCKETTE



ATAULFO saquea el reino de su hermano, el rey LEONARD, y deja en el castillo unos seres encantados para destruir al rey LEONARD e impedirle el paso a la sala de los tesoros.



Los malignos han esparcido las fórmulas secretas del gran mago Merlín, por todos los rincones de su inmenso castillo. ¿Podrás ayudarle a recuperarlas, para que pueda fabricar la piedra Filosofal?

VISITE LA DIVISION OnLine Galerías Preciados

A LA VENTA EN GALERIAS
Marcando estilo.

Y EN TODOS LOS DISTRIBUIDORES DE NUESTROS PRODUCTOS

Editado y distribuido en España por:

MIND GAMES ESPAÑA S.A.

Mariano Cubi, 4 Entlo. Tel. 218 34 00 - 08006 Barcelona



te ofrece el mejor programa de simulación de vuelo creado hasta la fecha.



MSX
AMSTRAD
COMMODORE

MSX
AMSTRAD
COMMODORE

El Spitfire 40 no es sólo el mejor simulador de vuelo de los aviones más famosos de todos los tiempos, también es una espectacular aventura de guerra.
La escena es el verano de 1940 y tú eres un piloto recién entrenado, destinado en un Escuadrón Spitfire en algún lugar del Sureste de Inglaterra. Como uno de tantos jóvenes en 1940, aprenderás que un Spitfire no es un avión corriente, descubrirás sus especiales capacidades y lo más importante de todo, cómo manejarlo durante el combate.

★★★★★★★★★★★★★★★★ Si están agotados en tu tienda habitual ¡¡LLAMANOS!! ★★★★★★★★★★★★★★★★

Si deseas información y participar en los importantes sorteos que ZAFICHIP celebrará durante el año. . . ¡ESCRIBENOS!



ZAFIRO SOFTWARE DIVISION
Paseo de la Castellana, 141. 28046 Madrid
Tel. 459 30 04. Tel. Barna. 209 33 65. Télex: 22690 ZAFIR E